Error-backpropagation in Networks of Fractionally Predictive Spiking Neurons

Sander M. Bohte

CWI, Life Sciences Group
Science Park 123, NL-1097XG Amsterdam, The Netherlands
sbohte@cwi.nl   http://www.cwi.nl/~sbohte

Abstract. We develop a learning rule for networks of spiking neurons where signals are encoded using fractionally predictive spike-coding. In this paradigm, neural output signals are encoded as a sum of shifted power-law kernels. Simple greedy thresholding can compute this encoding, and spike-trains are then exactly the signal's fractional derivative. Fractionally predictive spike-coding exploits natural statistics and is consistent with observed spike-rate adaptation in real neurons; its multiple-timescale properties also reconciles notions of spike-time coding and spike-rate coding. Previously, we argued that properly tuning the decoding kernel at receiving neurons can implement spectral filtering; the applicability to general temporal filtering was left open. Here, we present an error-backpropagation algorithm to learn decoding these filters, and we show that networks of fractionally predictive spiking neurons can then implement temporal filters such as delayed responses, delayed match-to-sampling, and temporal versions of the XOR problem.

1 Introduction

Real biological neurons compute in continuous time via the exchange of electrical pulses or *spikes*, and algorithmic descriptions of neural information processing in terms of spikes likely holds the key to resolving the scientific question of how biological spiking neurons work. The interest in the computational properties of spiking neurons was boosted in particular by findings from experimental and theoretical neuroscience [12, 16], which suggested that the precise timing of individual spikes can be important in neural information processing. This has led to great debate, as the general assumption in neuroscience has always been that it is the neuron's spike rate that encodes information.

The notion of neural spike-time coding has resulted in the development of a number of spiking neural network approaches demonstrating that spiking neurons can compute using precisely times spikes similar to traditional neurons in neural networks [14, 3, 13]. Still, in spite of the successes of spiking neural networks, and the theoretical appeal of spike-time coding, it has remained a challenge to extend spike-based coding to computations involving longer timescales. Recurrent spiking neural network approaches can achieve longer memory, though they are notoriously hard to train [21]; edge-of-stability dynamical systems methods like reservoir computing show promise [11, 4], although they require many neurons and spikes, and mostly disregard notions of spike-time coding.

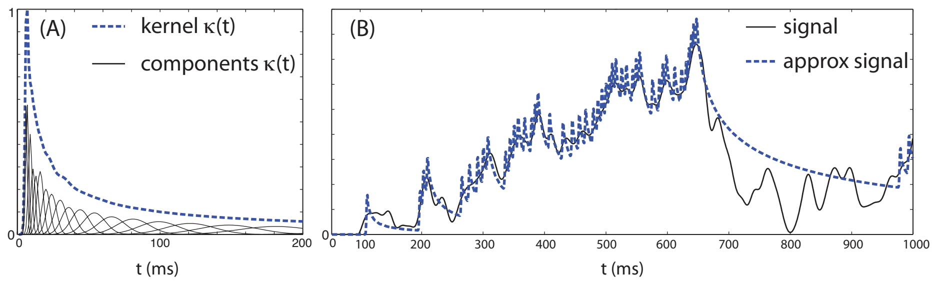

Fig. 1. (A) Power-law kernel (dashed blue) for $\beta = 0.8$, and Guassian components κ_k (B) Signal (black) approxiamated with a sum of power-law kernels (dashed blue).

Based on recent neuroscience findings [10], and reconciling the notions of both spike-time coding and spike-rate coding, we proposed a novel scheme for spike-based neural coding based on the observation that a mild derivative – a *fractional* derivative – of a signal can under certain conditions *be* a series of spikes [2]. In this framework, neural spiking is a statistically efficient means of encoding time-continuous signals. It does so by approximating the internally computed neural signal as sum of shifted kernels, where these kernels decay following a power-law (e.g. figure 1A). Power-law kernels provide much longer traces of past signals as compared to exponentially decaying kernels [6], and are thus much more suitable for computing temporal functions over behaviorally relevant timescales.

In this paper, we exploit key properties of the fractional-spike coding framework to learn functions over behaviorally relevant timescales. We capitalize on the fact that power-law kernels can be approximated for example using a weighted sum of exponential functions. We show that useful temporal filters can be learned by adapting these composite weights when decoding spike-trains *at the receiving synapse*. For this task, we derive error-backpropagation in the fractional spike-coding paradigm. With this learning rule we show that networks of fractionally predictive neurons can learn functions through time like delayed timer-functions and recall tasks like delayed-match-to-sample.

2 Fractionally Predictive Spiking Neurons

Starting from the standard Linear-nonlinear neuron model [1], an artificial neuron j computes an internal variable $y_j(t)$ as a function over the weighted sum of filtered inputs $x_j(t)$: $y_j(t) = \mathcal{F}(x_j(t))$ and $x_j(t) = \sum_{i \in \mathcal{J}} w_{ij} f_i(y_i(t))$, where $\mathcal{J}$ is the set of presynaptic neurons i to neuron j, and $f_i(y_i(t))$ denotes the (temporal) filter that computes $(f_i * y_i)(t)$.

As defined in [2], a fractionally predictive spiking neuron j approximates the internal signal $y_j(t)$ with $\hat{y}_j(t)$ as a sum of shifted power-law kernels centered at spike-times $\{t_i\}$:

$$y_j(t) \approx \hat{y}_j(t) = \sum_{t_j < t} \kappa(t - t_j).$$

The fractional derivative of order α of this approximation $\hat{y}_j(t)$ is just the spike-train $\{t_i\}$ when the kernel $\kappa(t)$ decays proportional to a power-law $\kappa(t) \propto t^{-\beta}$ when $\alpha = 1 - \beta$ [2]:

$$\frac{\partial^\alpha \hat{y}_j(t)}{\partial t^\alpha} = \sum_{t_j<t} \delta(t - t_j).$$

Such signal approximation $\hat{y}_j(t)$ can be achieved by computing the difference between the current signal estimation and the (emitted) future estimation (prediction) $\hat{y}(t)$, adding a spike t_i when this difference exceeds a threshold ϑ:

$$\begin{aligned} z(t) &= y(t) - \hat{y}(t) \\ t_i &= t \quad \text{if} \quad z(t) > \vartheta \end{aligned}$$

With a single, positive threshold only positive signal deviations are transmitted, and for negative deviations the transmitted signal decays as $t^{-\beta}$ (closely matching actual spiking neuron behavior [15]). Such signal approximation is shown in figure 1B, where the height of the kernel κ is set to two times the threshold ϑ. Alternatively, the signal approximation can be precise up to ϑ if we use positive and negative spikes to signal respectively positive and negative deviations, for instance using two tightly coupled spiking neurons [2].

We use the fact that a power-law kernel $\kappa(t)$ can be approximated as a sum (or cascade) of different, weighted exponentially decaying functions $\kappa_k(t)$ [6]:

$$\kappa(t) \approx \sum_k \kappa_k(t),$$

as illustrated in figure 1A. This lets us rewrite $\hat{y}_i(t)$ as a sum of components $y_i^k(t)$:

$$\hat{y}_i(t) = \sum_{t_i<t} \sum_k \kappa_k(t - t_i) = \sum_k y_i^k(t).$$

At a receiving neuron j, a temporal filter κ_{ij} of the signal $\hat{y}_j(t)$ from neuron j can then be created by weighing these components with weights w_{ij}^k at the receiving synapse:

$$\kappa_{ij}(t) = \sum_k w_{ij}^k y_i^k(t).$$

and the neuron's input is thus computed as:

$$x_j(t) = \sum_{i \in \mathcal{J}} \sum_{t_i<t} \kappa_{ij}(t - t_i) = \sum_{i \in \mathcal{J}} w_{ij}^k y_i^k(t),$$

Note that for $w_{ij}^k = w_{ij} \forall\, k$, the input at neuron j decodes a weighted version of the output of presynaptic neurons i.

3 Learning in Networks of Fractionally Predictive Spiking Neurons

We consider a standard fully connected feedforward neural network, with input layer $\mathcal{I}$, hidden layer $\mathcal{H}$, and output layer O, populated with neurons i, j and

m. We derive standard error-backpropagation learning rules for adjusting the components w_{ij}^k of the filtering kernels $\kappa_{ij}(t)$ for each connection in the network.

Given desired output activation pattern $s_k(t)$ for each output neuron k, we define a standard quadratic error measure in terms of the output $\hat{y}$:

$$E(t) = \sum_{m \in \mathcal{O}} \left(s_m(t) - \hat{y}_m(t)\right)^2$$

The goal is to adjust each weight w_{ij}^k (and w_{jm}^k) in the network so as to minimize the error over some time-period $[T, T']$:

$$\Delta w_{ij}^k \propto -\frac{\partial \sum_{t=T}^{T'} E(t)}{\partial w_{ij}^k} = \sum_{t=T}^{T'} \frac{\partial E(t)}{\partial w_{ij}^k},$$

(as the error-contributions are conditionally independent).

For the output layer, we have:

$$\frac{\partial E(t)}{\partial w_{jm}^k} = \frac{\partial E(t)}{\partial \hat{y}_m(t)} \frac{\partial \hat{y}_m(t)}{\partial x_m(t)} \frac{\partial x_m(t)}{\partial w_{jm}^k} = \left(s_m(t) - \hat{y}_m(t)\right) \mathcal{F}'(x_m(t)) y_j^k(t),$$

where $\mathcal{F}'(x_m(t))$ denotes the derivative $\partial \hat{y}_m / \partial x_m(t)$.

For weights in the hidden layer, the error-contributions become:

$$\begin{aligned} \frac{\partial E(t)}{\partial w_{ij}^k} &= \sum_{m \in \mathcal{O}} \frac{\partial E(t)_m}{\partial \hat{y}_m(t)} \frac{\partial \hat{y}_m(t)}{\partial x_m(t)} \sum_{k'} \frac{\partial x_m(t)}{\partial y_j^k(t)} \frac{\partial y_j^k(t)}{\partial \hat{y}_j(t)} \frac{\partial \hat{y}_j(t)}{\partial x_j(t)} \frac{\partial x_j(t)}{\partial w_{ij}^k} \\ &= \sum_{m \in \mathcal{O}} \left[\left(s_m(t) - y_m(t)\right) \mathcal{F}'(x_m(t)) \sum_{k'} w_{jm}^{k'} \right] \mathcal{F}'(x_j(t)) y_i^k(t). \end{aligned}$$

Here, we take the transfer-function $\hat{y} = \mathcal{F}(x(t))$ to be piece-wise linear, with $\mathcal{F}(x(t)) = 0$ for $x(t) < 0$, and $\mathcal{F}(x(t)) \approx \alpha x(t)$ otherwise; a lack of input signals then automatically maps to a lack of output signals.

4 Experiments

We illustrate the efficacy of the derived error-backpropagation learning rule with some examples of behaviorally relevant temporal computations. Given a defined input-output relationship, we computed the respective approximation $\hat{y}(t)$ with power-law kernels, obtaining corresponding input-output spike-trains. We trained the network to minimize the error between the actual output and the desired output, both in terms of the respective power-law kernel approximation.

In all experiments, we use a power-law kernel with $\beta = 0.8$ (after [10]), which we approximate with 40 Gaussians, with centers μ_k distributed evenly over the log of the timeframe [5, 10000]ms, increasing variance σ_k as $\mu_k/5$. We used a greedy search heuristic to find the weighing of the individual gaussians such that their sum closely approximated the desired power-law decay. The learning rate was set at 0.01, and we considered the error over a time range of $(0, 2000]$ms (as

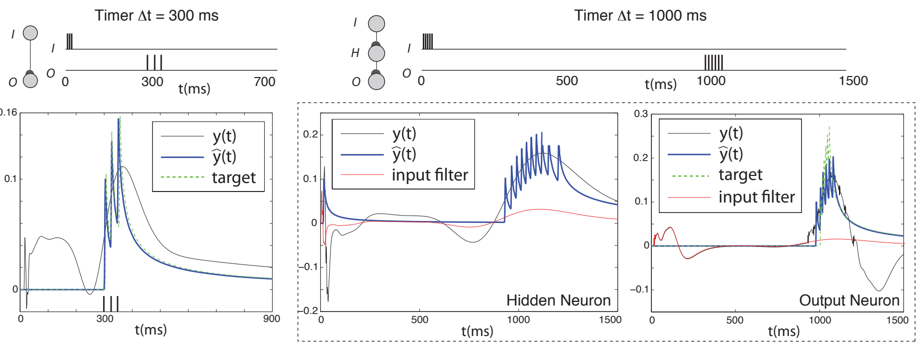

Fig. 2. Neural timers. Left: learning a 300ms delay directly from input. Top: input-target spike pattern. Bottom: Learned response after 400 epochs, with the kernel approximation $\hat{y}(t)$ matching the target output. Target (green dashed) and internal signal $y(t)$ (solid black) and signal kernel approximation $\hat{y}(t)$ (dashed blue). Right: learning a larger response at $\Delta t = 1000$ms, with an additional hidden neuron. Top: input-target spike pattern. Bottom: Learned response after 900 epochs, for the hidden neuron (center) and the output neuron (right). Shown in red is the developed temporal filter.

all relevant patterns were defined within this range). It should be noted that in both hidden and output layer, the signal-approximation incurs a delay on the computed signal approximation equal to the rise-time of the kernel; in all figures (and in the error-computations), we subtracted this fixed delay for clarity. Setting the maximum kernel height $\kappa_M = 0.1$, we used a threshold $\vartheta = 0.5\kappa_M$. Weights were initialized such that input elicited some spikes in each hidden neuron. We trained the networks until the error was less than one “misplaced” spike.

Spiking neuron as timer. In figure 2, we show how a single fractionally predictive neuron can learn to give a precise delayed response. Given three input spikes at the start of the trial, the neuron responds with three carefully timed spikes some 300ms later. Such a direct input-output relation however was not sufficient to learn a precise mapping for substantially larger delays, such as 1,000ms. In the center and right figure, we show how the addition of a hidden neuron can help the output neuron to learn a delayed precise response. The hidden neuron computes a broad delay of about 900ms, and the output neuron then derives a much more precise response. Shown in red are also the effective temporal input filters that the respective neurons develop.

Delayed Match To Sample. Many behavioral tasks in some way require an animal to remember something it had experienced earlier: delayed match-to-sample tasks. We taught a simple version of this task. One of two input neurons emits a few spikes, and when the same input neuron spikes some 200ms later, the output neuron has to respond with an additional 200ms delay (see insets in figures 3A-D); if the other input neuron spikes, the output neuron has to remain silent. A network using four hidden neurons can successfully learns this task (figure 3), also for variations with more spikes and longer delays (not shown).

Delayed XOR. We were also able to train the network on various delayed temporal XOR problems similar to those in [3] (not shown for lack of space).

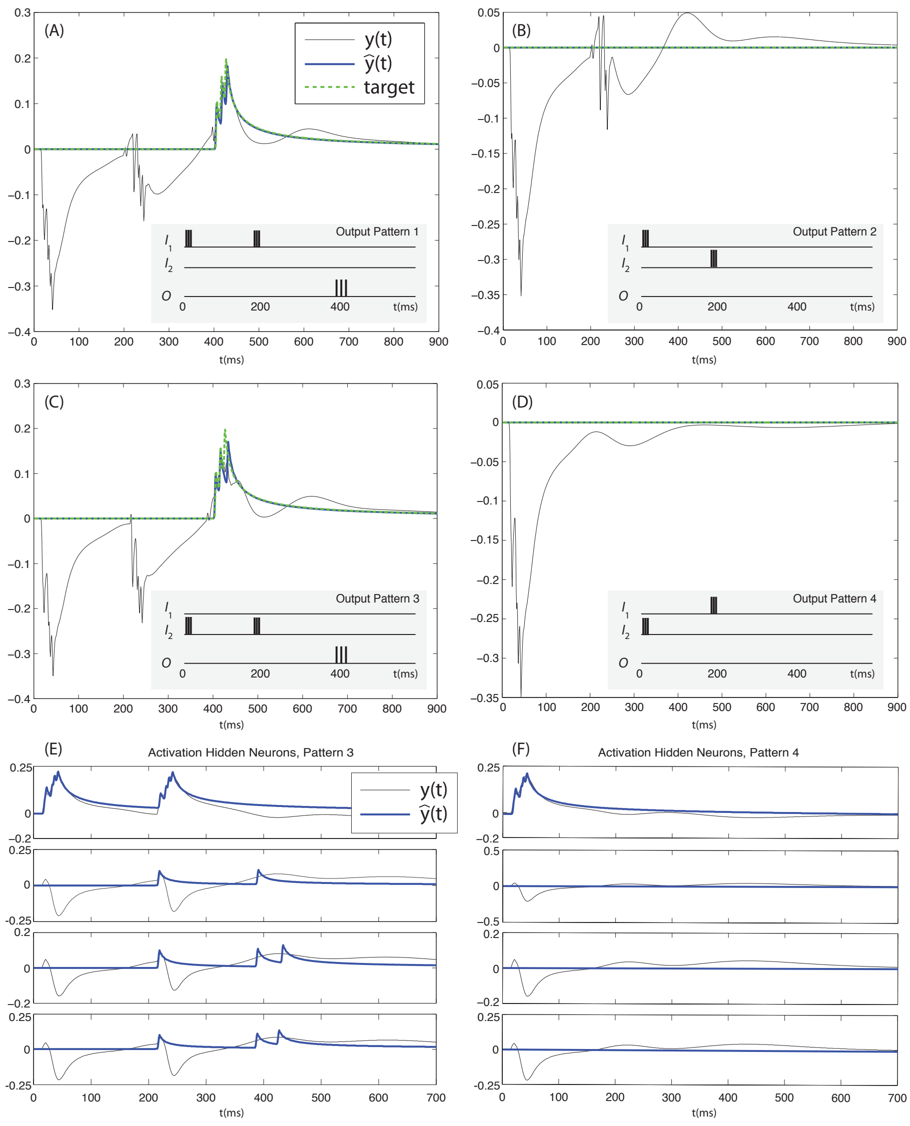

Fig. 3. Learning delayed match-to-sample. (A)-(D): the output neuron in a 2-4-1 network correctly learns the spike-responses $\hat{y}(t)$ to the four input-output patterns (inset boxes), in about 2000 epochs. (E)-(F): hidden neuron responses for patterns 3 and 4.

5 Discussion

We developed a learning algorithm for supervised learning in networks of spiking neurons using the fractionally predictive spike-coding paradigm from [2]. This coding paradigm allows for a natural signal encoding over multiple timescales, consistent with the self-similar statistical properties of natural signals [22]. It also allows for a natural reconciliation of spike-time coding and spike-rate coding as expressions of fractionally predictive spike-coding at different timescales. Within this paradigm, decoding of the fractional spike-code amounts to summing power-law kernels. As we noted in [2], such decoding allows for straightforward spectral filtering, as such power-law kernels can be decomposed as a sum of exponential kernels with different time-constants; spectral filtering is achieved by decoding only certain parts of such composite kernels. The open question was whether this could also be exploited for learning temporal patterns.

The derived error-backpropagation algorithm shows that temporal filters can be learned in networks of fractionally predictive spiking neurons, for a number of tasks, over behaviorally relevant time-courses. Effectively, the exponentials that can be composed to a power-law kernel are used as time-delays. Such delays have been explored before in the context of spike-time coding [14, 3, 18], but only over small time-courses (tens of milliseconds), consistent with biologically observed axonal delays between neurons. Behaviorally relevant tasks operate typically over substantially longer timescales; and it is interesting to note that models of reinforcement learning [20, 9] in fact similarly employ neurons that are the an array of exponentials to allow standard neural networks to learn sequential tasks. Izhikevich furthermore showed that resonance-like input-responses can be modeled within the standard Hodgkin-Huxley neuron [8].

Neural coding over longer timescales is implicit in the fractionally predictive spike-coding paradigm, and corresponds at the encoding side to the physiologically well-established phenomenon of spike-rate adaptation. Our learning rule effectively conjectures that similar multiple-timescale machinery is effective *and tunable* at the receiving part of individual synapses. To some degree this must be true, as real neurons often have complex temporal receptive field dynamics, and many neurons exhibit sustained activity in response to a brief activation [7].

The learning rule presented here allows for learning input-output relations over different timescales: at short timescales, we can exactly learn relative spike-times; at longer timescales, we are no longer able to learn exact spike-times but instead learn approximate instantaneous spike-rates.

We used a single positive threshold spiking neuron model to approximate the internally computed signal into a sum of power-law kernels. This arrangement by necessity cannot encode signal decreases that are faster than the decay of the sum of power-law kernels. This could be remedied by arranging two neurons such that the signal is effectively approximated using both a positive and a negative threshold, as we did in [2]. We did not use such an arrangement here for three reasons, the first being simplicity. Secondly, power-law decay of a signal is, under certain conditions, consistent with an optimal Bayesian observer model [19], and thirdly the corresponding asymmetric detection of signal changes up versus down is well known in the psychophysical literature [5].

Having a standard supervised learning algorithm will allow us to further explore the fractionally predictive spike-coding paradigm in the context of recurrent neural networks, in particular reservoir computing. We are also working on developing the paradigm for reinforcement learning, as that is how most animals learn, and, as noted, that is also where the closest comparable modeling work has already taken place. Importantly, in the latter paradigm, methods like attention-gated reinforcement learning on average compute the error-backpropagation gradient [17], suggesting a direct link to the presented work.

References

1. Bishop, C.: Neural networks for pattern recognition. Oxford Univ. Press (1995)
2. Bohte, S., Rombouts, J.: Fractionally predictive spiking neurons. In: Lafferty, J., Williams, C.K.I., Shawe-Taylor, J., Zemel, R., Culotta, A. (eds.) Advances in Neural Information Processing Systems 23, pp. 253–261 (2010)
3. Bohte, S., Kok, J., La Poutre, H.: Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing 48(1-4), 17–37 (2002)
4. Buesing, L., Schrauwen, B., Legenstein, R.: Connectivity, dynamics and memory in reservoir computing with binary and analog neurons. Neural Comp. (In Press)
5. DeWeese, M., Zador, A.: Asymmetric dynamics in optimal variance adaptation. Neural Computation 10(5), 1179–1202 (1998)
6. Drew, P., Abbott, L.: Models and properties of power-law adaptation in neural systems. Journal of neurophysiology 96(2), 826 (2006)
7. Gnadt, J., Andersen, R.: Memory related motor planning activity in posterior parietal cortex of macaque. Experimental Brain Research 70(1), 216–220 (1988)
8. Izhikevich, E.: Resonate-and-fire neurons. Neural networks 14(6-7), 883–894 (2001)
9. Ludvig, E., Sutton, R., Kehoe, E.: Stimulus representation and the timing of reward-prediction errors in models of the dopamine system. Neural Computation 20(12), 3034–3054 (2008)
10. Lundstrom, B., Higgs, M., Spain, W., Fairhall, A.: Fractional differentiation by neocortical pyramidal neurons. Nature neuroscience 11(11), 1335–1342 (2008)
11. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: A new framework for neural computation based on perturbations. Neural Computation 14(11), 2531–2560 (2002)
12. Markram, H., Tsodyks, M.: Redistribution of synaptic efficacy between neocortical pyramidal neurons. Nature 382(6594), 807–810 (1996)
13. McKennoch, S., Voegtlin, T., Bushnell, L.: Spike-timing error backpropagation in theta neuron networks. Neural computation 21(1), 9–45 (2009)
14. Natschläger, T., Ruf, B.: Spatial and temporal pattern analysis via spiking neurons. Network: Computation in Neural Systems 9(3), 319–332 (1998)
15. Pozzorini, C., Naud, R., Mensi, S., Gerstner, W.: Multiple timescales of adaptation in single neuron models. In: Front. Comput. Neurosci. Conference Abstract: Bernstein Conference on Computational Neuroscience (2010)
16. Rieke, F., Warland, D., Bialek, W.: Spikes: exploring the neural code (1999)
17. Roelfsema, P., Van Ooyen, A.: Attention-gated reinforcement learning of internal representations for classification. Neural Computation 17(10), 2176–2214 (2005)
18. Schrauwen, B., Van Campenhout, J.: Extending spikeprop. In: Proceedings IJCNN 2004. vol. 1. IEEE (2005)
19. Snippe, H., van Hateren, J.: Recovery from contrast adaptation matches ideal-observer predictions. JOSA A 20(7), 1321–1330 (2003)
20. Suri, R., Schultz, W.: Learning of sequential movements by neural network model with dopamine-like reinforcement signal. Exp. Brain Res 121(3), 350–354 (1998)
21. Tino, P., Mills, A.: Learning beyond finite memory in recurrent networks of spiking neurons. Neural computation 18(3), 591–613 (2006)
22. Van Hateren, J.: Processing of natural time series of intensities by the visual system of the blowfly. Vision Research 37(23), 3407–3416 (1997)