Simulating the phototaxis response in C. Elegans

Clemens Hutter
chutter [at] uos.de

October 18, 2016

Abstract

The nematode C. elegans will move back once light is shown on its head. This behaviour know as phototaxis was quantified by Ward et al. [1]. They looked at the timing (onset and duration) of the backward movement after light stimuli of different intensity and wavelength are applied to the worm. With the final goal to replicate the timings for all nine different stimulus configurations tested in the paper I started out with the arbitrary specific case of 350 nm and an intensity of $-1.73 \cdot log(\frac{I}{I_O})$. In this case the backwards movement starts roughly 1.7 seconds after stimulus onset and will last for about 7 seconds [1].

1 Methodology

I simulated the activity after the stimulus in a simplified neural network [2] based on the C. elegans connectome and used an evolutionary algorithm to arrive at parameters that replicate the response timing in two target motor neurons. The full source code is available here: https://github.com/rauwuckl/CElegansPhototaxis

1.1 Network

The network (Figure 1) consists of 4 sensory neurons, 4 interneurons and 2 motor neurons [2]. All of the neurons are modelled as Izhikevich quadratic integrate-and-fire spiking neural models [3]. (with a = 0.02, b= 0.2, c= -65, d= 6). Additionally the 4 sensory neurons receive an additive input current related to surrounding light as

$$I_{light} = \frac{|750 - \lambda| \cdot (30 + i)}{\lambda}$$

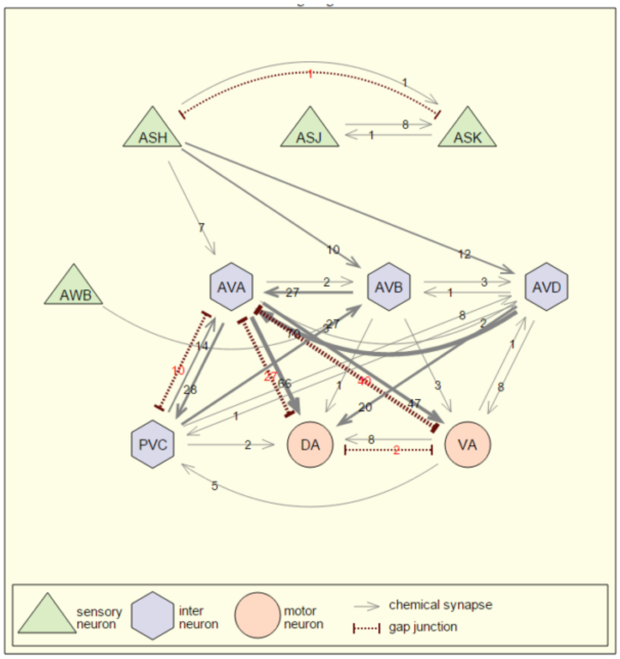


Figure 1: Full inter neural connectivity for the sensory, interneuron and muscle neurons used in our phototaxis model. IV.

where λ is the wavelength of light (in this case 350 nm) and i is the intensity (here $10^{-1.73} \cdot 20 = 0.37$) [2].

Chemical synapses are modelled as Instantaneous Rise and Single-Exponential Decay synapses:

$$\frac{dg}{dt} = -\frac{1}{\tau} \cdot g$$
$$I_{syn} = w \cdot g \cdot (V_{reversal} - V_{post})$$

where g is the exponentially decaying conductivity which is set to 1 at each arriving presynaptic spike, τ is a time constant fixed for all synapses, w is a synaptic strength constant (different value for each synapse), $V_{reversal}$ is the reversal potential ($-70mV$ for inhibitory and $0mV$ for excitatory synapses), V_{post} is the current membrane potential in the postsynaptic neuron and I_{syn} the synaptic current that will be added to the postsynaptic neuron ($g(t) = w \cdot e^{-(t-t_0)/\tau}$ is an equivalent formulation). If there are connections between 2 neurons going back and forth (eg. ASJ→ASK and ASK→ASJ) they are explicitly modelled as 2 synapses.

Electrical synapses are modelled symmetrical as

$$I_{syn} = w \cdot (V_{pre} - V_{post})$$

where w is again the strength for a given synapse and V_{pre}/V_{post} the membrane potential in the presynaptic/postsynaptic neuron.

There are 27 chemical and 5 electrical connections (see arrows in Figure 1). Each of these connections is formed by different numbers of actual synapses (see numbers in the Figure 1). Since we are assuming a global value for τ we can (for a given connection between two neurons) just add up all the synaptic currents into one combined synaptic model. The strength of theses 'virtual' synapses (simply synapse from now on) will then be proportional to the number of actual synapses (numbers in Figure 1). i.e. we have only one 'virtual' synapse instead of 12 actual synapses from ASH to AVD but the w for this synapse will be 12 times stronger.
Therefore our parameter space consists of 32 synaptic weights plus τ. These are represented in a 33-dimensional vector $\vec{w}$ with all elements between 0 and 1. Each value of the parametervector is then mapped into a reasonable range for the corresponding parameter:

$$\begin{aligned} w_i^{syn} &= |\vec{w}_i - 0.5| \cdot N_i^{synapses} \cdot 0.5 \\ w_j^{electrical} &= (\vec{w}_j \cdot N_j^{synapses}) \in [0;1] \\ \tau &= \vec{w}_{33} \cdot 19 + 0.1 \end{aligned}$$

Chemical synapses are set to excitatory for $\vec{w}_i \geq 0.5$ and inhibitory otherwise. Weights for electrical synapses are clipped such that they do not exceed [0:1].

The network is implemented for the brian2 simulator in Python.

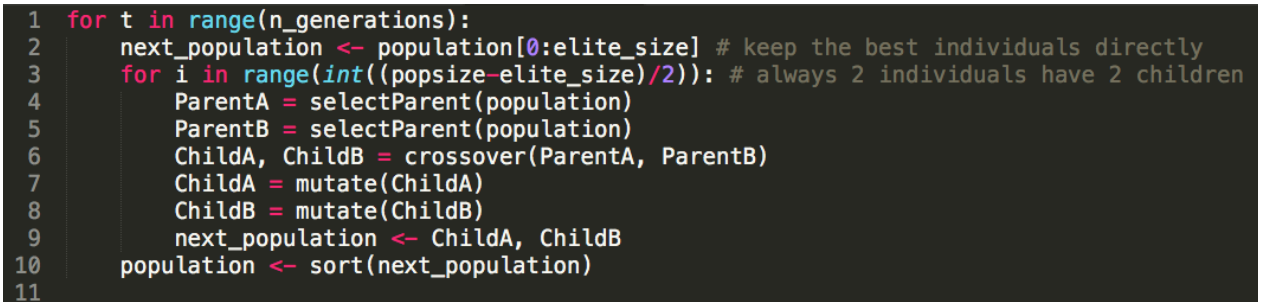
1.2 Parameter finding with evolutionary algorithms

I was looking for spike patterns in the motor neurons DA and VA that could plausibly cause backwards movement starting 1.7 seconds after stimulus onset and lasting for 7 seconds [1]. I argue that this would require them to start spiking before the start of the actual movement and to spike for approximately the same duration as the duration of the backward movement. Furthermore the spiking should be regular. The fitness function derived from these constrains is:

$$\begin{aligned} differenceStart &= max((FirstSpike - StartResponse), 0)^2 \\ differenceEnd &= max((LastSpike - EndResponse), 0)^2 \\ differenceDuration &= ((LastSpike - FirstSpike) - DurationResponse)^2 \\ variance &= \text{variance of interspike intervals} \\ fitness &= -(20 \cdot differeceStart + 20 \cdot differenceEnd \\ &\quad + 20 \cdot differenceDuration + 25 \cdot variance) \end{aligned}$$

where the first two terms will only be non-zero if the spiking occurs after the response. The fitness is assessed for DA and VA independently in this way and then summed.

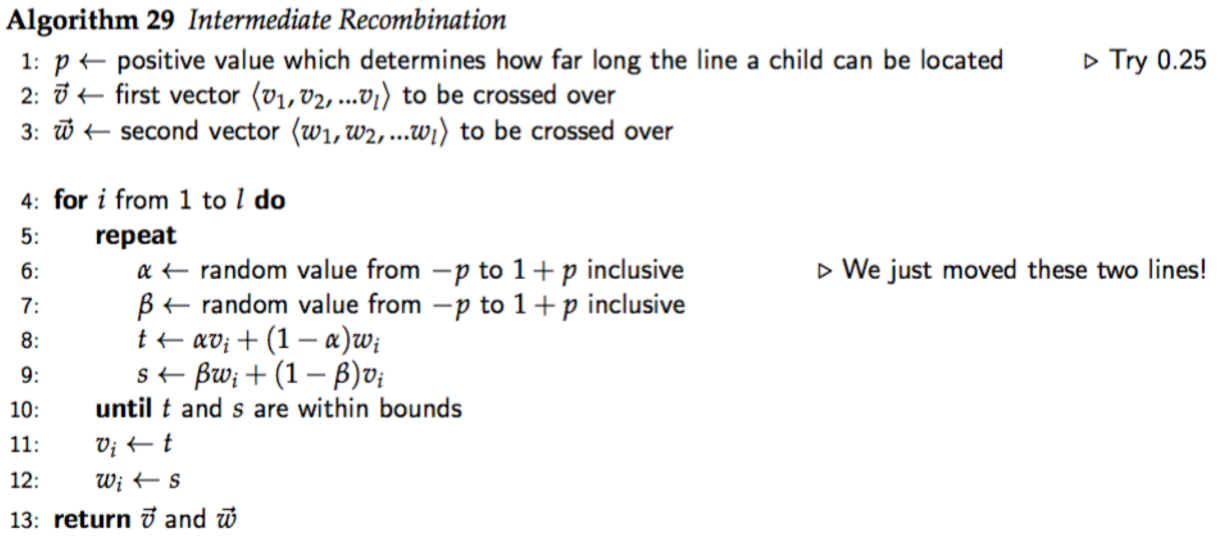
This fitness function is optimized by the following evolutionary algorithm (adapted from [4]):



```
for t in range(n_generations):
    next_population <- population[0:elite_size] # keep the best individuals directly
    for i in range(int((popsize-elite_size)/2)): # always 2 individuals have 2 children
        ParentA = selectParent(population)
        ParentB = selectParent(population)
        ChildA, ChildB = crossover(ParentA, ParentB)
        ChildA = mutate(ChildA)
        ChildB = mutate(ChildB)
        next_population <- ChildA, ChildB
    population <- sort(next_population)
```

Figure 2: Pseudocode for the evolutionary algorithm

selectParent draws 2 random individuals from the population with replacement and selects the fitter of the two. The **mutate** function simply adds independent Gaussian noise to each element of $\vec{w}$ ($\sigma = 0.001$). The **crossover** function performs *Intermediate (Line) Recombination* to produce two children based on two based on two parents:



Algorithm 29 *Intermediate Recombination*

```
p ← positive value which determines how far long the line a child can be located    ▷ Try 0.25
$\vec{v}$ ← first vector $\langle v_1, v_2, ... v_l \rangle$ to be crossed over
$\vec{w}$ ← second vector $\langle w_1, w_2, ... w_l \rangle$ to be crossed over

for $i$ from 1 to $l$ do
    repeat
        $\alpha$ ← random value from $-p$ to $1 + p$ inclusive    ▷ We just moved these two lines!
        $\beta$ ← random value from $-p$ to $1 + p$ inclusive
        $t \leftarrow \alpha v_i + (1 - \alpha) w_i$
        $s \leftarrow \beta w_i + (1 - \beta) v_i$
    until $t$ and $s$ are within bounds
    $v_i \leftarrow t$
    $w_i \leftarrow s$
return $\vec{v}$ and $\vec{w}$
```

Figure 3: Pseudocode for Intermidiate Recombination [4, p. 42]

The algorithm is implemented in Python and currently employs a multiprocess architecture to make full use of the strong PC available at the lab. To achieve this the entire population is divided into multiple subsets and the fitness in each subset is evaluated by a different job on a different CPU (Data Parallelism). For machines with only 1 or 2 cores a single threaded architecture might be faster. This can be easily changed in the source code.

2 Results

If no noise is added to the membrane potential of the neurons a very good fit is quite easily achievable. But real neurons will undoubtedly be subject to some noise. In this case the quality of a given parameter set $\vec{w}$ will vary considerably from run to run. So the fitness function for a given individual (parameter set) is extremely stochastic.

For this summery I left the evolution running for 82 generations, while evaluating the fitness for each individual at each generation 4 times and averaging over it to achieve a somewhat reliable value. Then I simulated the network for the fittest individual nine times to give a rough idea of the different activity patterns that arise.

- In the best case scenario the network very precisely replicated the response timing (Figure 4).
- The most common outcome however can be seen in Figure 5. Here the network starts spiking but does not calm back down to rest within the simulation time of 10.5 seconds.
- Also common is that the response will last exactly as long as the stimulus exiting the network (Figure 6).

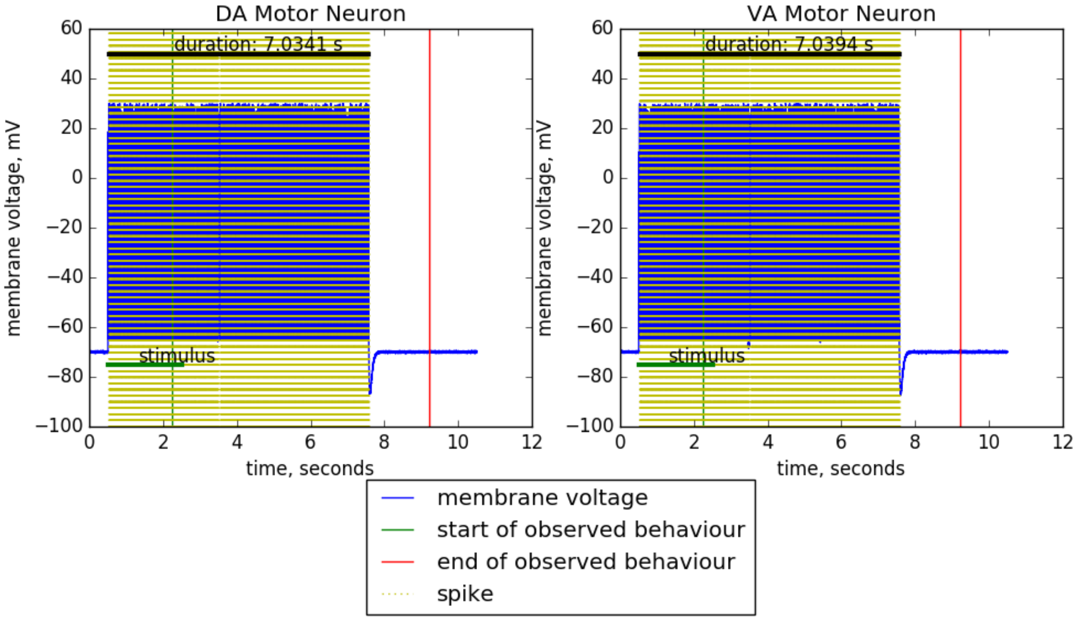


Figure 4: Trace of optimal run. (3 out of 9 runs with response durations of 7.0, 7.1 and 6.0 seconds)

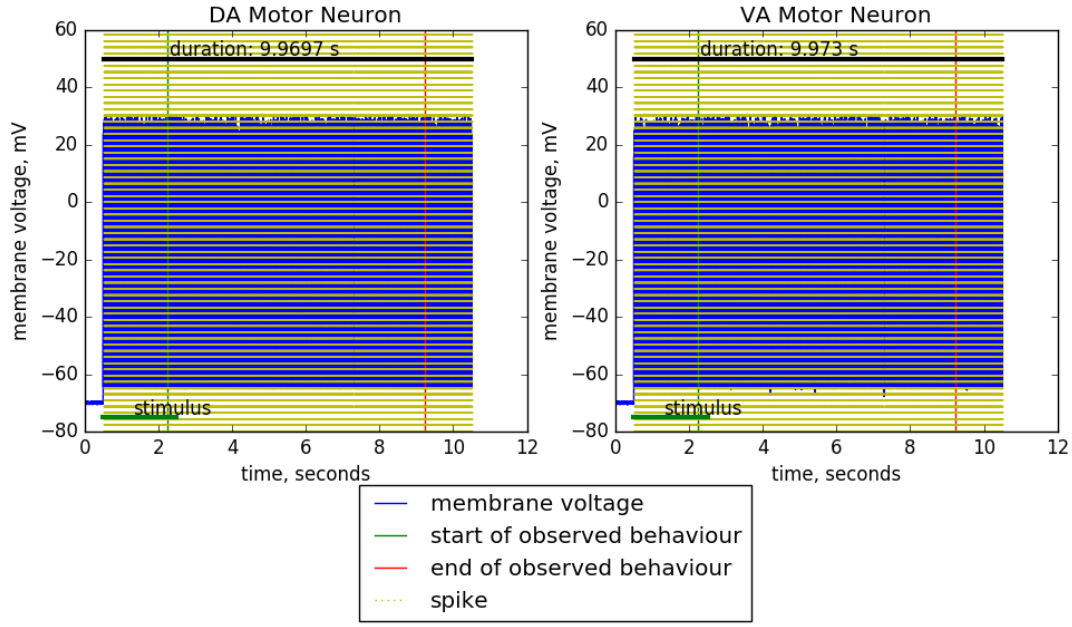


Figure 5: Most common trace. Spiking does not stop. (5 out of 9 runs)

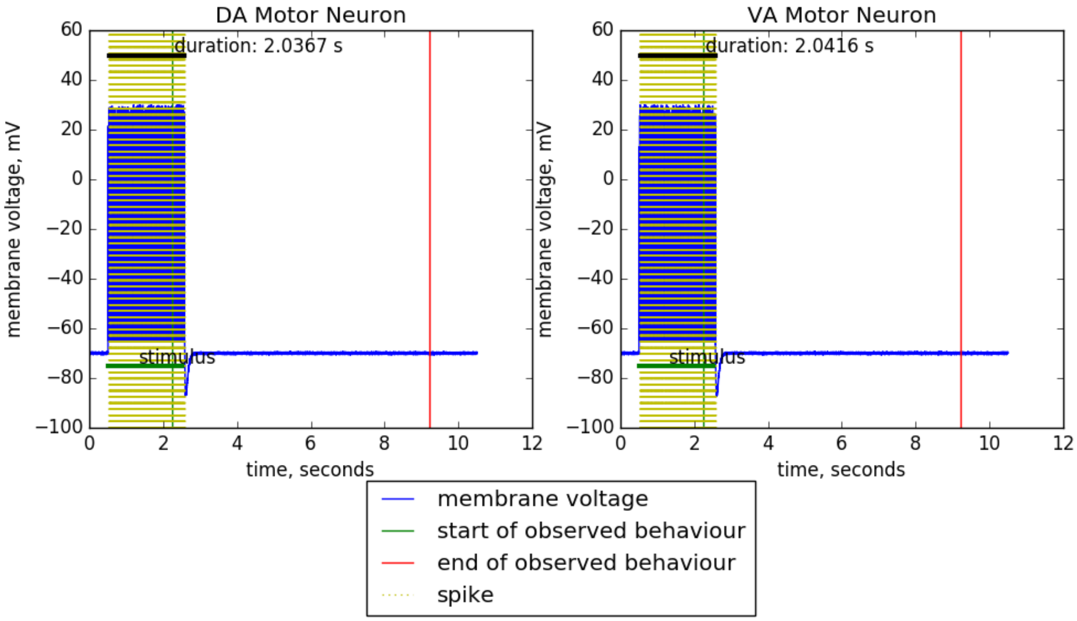


Figure 6: Spiking stops immediately. (1 out of 9 runs)

3 Discussion

I believe that the target response we are trying to achieve within the network is very unstable. The network has to get excited by external input which then is turned off after which the spiking response must not change. Then it has to stay excited for a set amount of time and after that it autonomously has to calm down without any change in input.
This I think happens once a short interval occurs in which all neurons in the highly recursive network are *'by chance'* in their refractory period. Then the self sustaining circle of excitation is stopped and the activity fades away. But as soon as we add even a little noise the exact spike times of all neurons will be shifted a little against each other. The interval of accidental silence can now occur at a completely different time (or not at all). Because of this intuition I figure that it is impossible to find parameters which are able to reproduce the behaviour reliable over multiple runs.

Also it is still unclear how higher activity in those two motorneurons actually relates to backward movement.

Disclaimer: *What is referred to as neurons (DMA, AVA, etc.) in this paper are actually neuron classes.*

References

[1] Alex Ward, Jie Liu, Zhaoyang Feng, and X Z Shawn Xu. Light-sensitive neurons and channels mediate phototaxis in C. elegans. *Nat Neurosci*, 11(8):916–922, aug 2008.

[2] Kofi Appiah, Pedro Machado, Alicia Costalago Meruelo, and T M Mcginnity. C . elegans Behavioural Response Germane to Hardware Modelling.

[3] Eugene M. Izhikevich. Simple model of spiking neurons. *IEEE Trans. Neural Networks*, pages 1569–1572, 2003.

[4] Sean Luke. *Essentials of Metaheuristics*. Lulu, second edition, 2013. Available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/.