

RESEARCH

Open Access

Nodal computation approximations in asynchronous cognitive models

James K Peterson

Abstract

Background: We are interested in an asynchronous graph based model, $\mathcal{G}(\mathbf{N}, \mathbf{E})$ of cognition or cognitive dysfunction, where the nodes \mathbf{N} provide computation at the neuron level and the edges $\mathbf{E}_{i \rightarrow j}$ between nodes \mathbf{N}_i and node \mathbf{N}_j specify internode calculation.

Methods: We discuss how to improve update and evaluation needs for fast calculation using approximations of neural processing for first and second messenger systems as well as the axonal pulse of a neuron.

Results: These approximations give rise to a low memory footprint profile for implementation on multicore platforms using functional programming languages such as **Erlang**, **Clojure** and **Haskell** when we have no shared memory and all states are immutable.

Conclusions: The implementation of cognitive models using these tools on such platforms will allow the possibility of fully realizable lesion and longitudinal studies.

Keywords: Cognition models; Graphs of computational nodes; Nodal computation approximation

Background

We note all brain models connect computational nodes (typically neurons) to other computational nodes using edges between the nodes. The proper background and motivation for this approach are discussed in (Peterson 2014a). That review article provides a reasonable introduction to the blend of ideas from mathematics, biophysics and signaling theory such as first and second messengers along with neurobiological concepts and computational implementation issues that are the foundation for the work presented here. We intend this article for researchers from computational, biological and mathematical backgrounds and the review provides an overview and a common set of notations and language. On the basis of that background material, we therefore model the neural circuitry of a brain using a directed graph architecture $\mathcal{G}(\mathbf{N}, \mathbf{E})$ consisting of computational nodes \mathbf{N} and edge functions \mathbf{E} which mediate the transfer of information between two nodes. Hence, if \mathbf{N}_i and \mathbf{N}_j are two computational nodes, then $\mathbf{E}_{i \rightarrow j}$ would be the corresponding edge function that handles information

transfer from node \mathbf{N}_i and node \mathbf{N}_j . We organize the directed graph using interactions between neural modules (visual cortex, thalamus etc) which are themselves subgraphs of the entire circuit. Once a directed graph is chosen to represent neural circuitry, the addition of new neural modules is easily handled as a subgraph addition. Although connectivity is time dependent, we can think of a useful brain model as a sequence of such graphs of nodes and edges. For simulation purposes, this means there is a finite sequence of times, $\{t_1, t_2, \dots, t_n\}$ and associated graphs $\mathcal{G}_i(\mathbf{N}_i, \mathbf{E}_i)$, where the subscript i denotes the time point t_i . In between these times, the graph has fixed connectivity and we can use a variety of tools to train the graph to meet input to output objectives. At each time point, we can therefore build our cognitive model using many different approaches as detailed in, e.g. (Friston 2005, 2010; Russo and Nestler 2013; Sherman 2004) and (Maia and Frank 2011). Indeed, some topology considerations for schizophrenia are directly addressed in (Li et al. 2012). However, here we are concerned with how to approximate the details of nodal processing so that computations can be performed efficiently with low memory footprint using the no shared memory model of a functional programming language such as **Erlang**, **Haskell** and **Clojure**. These languages allow us to design

Correspondence: petersj@clemson.edu
Department of Biological Sciences, Department of Mathematical Sciences,
Clemson University, Martin Hall O-304, BOX 340975, 340975 Clemson, SC, USA

simple processes that interact and have no shared memory state. On even a typical laptop today, we can have four cores and 16 GB of memory, so it is easily possible to simulate the interactions of many thousands of simple interacting neurons as simple processes. Recall in the graph context, the update equations for a given node then are given as an input/output pair. For the node N_i , let y_i and Y_i denote the input and output from the node, respectively. Then we have

$$\begin{aligned} y_i(t+1) &= I_i + \sum_{j \in \mathcal{B}(i)} E_{j \rightarrow i}(t) Y_j(t) \text{ where } Y_i(t+1) \\ &= \sigma_i(t)(y_i(t)). \end{aligned}$$

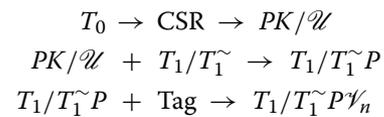
where I_i is a possible external input, $\mathcal{B}(i)$ is the list of nodes which connect to the input side of node N_i and $\sigma_i(t)$ is the function which processes the inputs to the node into outputs. This processing function is mutable over time t because second messenger systems are altering how information is processing each time tick. Hence, our model consists of a graph \mathcal{G} which captures the connectivity or topology of the brain model on top of which is laid the instructions for information processing via the time dependent node and edge processing functions. A simple look at edge processing shows the nodal output which is perhaps an action potential which is transferred without change to a synaptic connection where it initiates a spike in Ca^{+2} ions which results in neurotransmitter release. The efficacy of this release depends on many things, but we can focus on four: $r_u(i, j)$, the rate of re-uptake of neurotransmitter in the connection between node N_i and node N_j ; the neurotransmitter is destroyed via an appropriate oxidase at the rate $r_d(i, j)$; the rate of neurotransmitter release, $r_r(i, j)$ and the density of the neurotransmitter receptor, $n_d(i, j)$. The triple $(r_u(i, j), r_d(i, j), r_r(i, j)) \equiv T(i, j)$ determines a net increase or decrease of neurotransmitter concentration between the two nodes: $r_r(i, j) - r_u(i, j) - r_d(i, j) \equiv r_{net}(i, j)$. The efficacy of a connection between nodes is then proportional to the product $W_{ij} = r_{net}(i, j) \times n_d(i, j)$. Hence, each triple is a determining signature for a given neurotransmitter and the effectiveness of the neurotransmitter is proportional to the new neurotransmitter flow times the available receptor density. A very simple version of this is to simply assign the value of the edge processing function $E_{j \rightarrow i}$ to be the weight W_{ij} as is standard in a simple connectionist architecture. We want to be more sophisticated than this and therefore want to allow our nodal processing functions to approximate the effects of both first and second messenger systems; consult (Peterson 2014a) for more detail.

Cellular triggers

Now consider the transcriptional control of a regulatory molecule such as $NFKB$ (which plays a role in immune

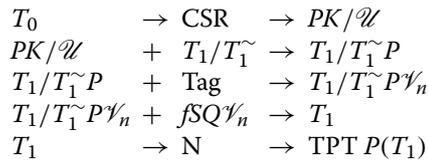
system response) or a neurotransmitter. We can call this a trigger and denote it by T_0 . This mechanism is discussed in a semi-abstract way in (Gerhart and Kirschner 1997); we will discuss even more abstractly. Consider a trigger T_0 which activates a cell surface receptor. Inside the cell, there are always protein kinases that can be activated in a variety of ways. Here we denote a protein kinase by the symbol PK . A common mechanism for such an activation is to add to PK another protein subunit \mathcal{U} to form the complex PK/\mathcal{U} . This chain of events looks like this: $T_0 \rightarrow CSR \rightarrow PK/\mathcal{U}$ where CSR denotes a *cell surface receptor*. PK/\mathcal{U} then acts to *phosphorylate* another protein. The cell is filled with large amounts of a transcription factor we will denote by T_1 and an inhibitory protein for T_1 we label as T_1^\sim . This symbol, T_1^\sim , denotes the *complement* or *anti* version of T_1 . In the cell, T_1 and T_1^\sim are generally joined together in a complex denoted by T_1/T_1^\sim . The addition of T_1^\sim to T_1 prevents T_1 from being able to access the genome in the nucleus to transcribe its target protein.

The trigger T_0 activates our protein kinase PK to PK/\mathcal{U} . The activated PK/\mathcal{U} is used to add a phosphate to T_1^\sim . This is called phosphorylation. Hence, $PK/\mathcal{U} + T_1^\sim \rightarrow T_1^\sim P$ where $T_1^\sim P$ denotes the phosphorylated version of T_1^\sim . Since T_1 is bound into the complex T_1/T_1^\sim , we actually have $PK/\mathcal{U} + T_1/T_1^\sim \rightarrow T_1/T_1^\sim P$. In the cell, there is always present a collection of proteins which tend to bond with the phosphorylated form $T_1^\sim P$. Such a system is called a **tagging** system. The protein used by the tagging system is denoted by \mathcal{V} and usually a chain of n such \mathcal{V} proteins is glued together to form a polymer \mathcal{V}_n . The tagging system creates the new complex $T_1/T_1^\sim P \mathcal{V}_n$. This gives the following event tree at this point:



Also, inside the cell, the tagging system coexists with a complimentary system whose function is to destroy or remove the tagged complexes. Hence, the combined system $\text{Tag} \leftrightarrow \text{Remove} \rightarrow T_1/T_1^\sim P$ is a **regulatory** mechanism which allows the transcription factor T_1 to be freed from its bound state T_1/T_1^\sim so that it can perform its function of protein transcription in the genome. The removal system is specific to \mathcal{V}_n molecules; hence although it functions on $T_1^\sim P \mathcal{V}_n$, it would work just as well on $Q \mathcal{V}_n$ where Q is any other tagged protein. We will denote the removal system which destroys \mathcal{V}_n tagged proteins Q from a substrate S by the symbol $fSQ \mathcal{V}_n$. This symbol means the system **acts** on $SQ \mathcal{V}_n$ units and outputs S via mechanism f . Note the details of the mechanism f are largely irrelevant here. Thus, we have the reaction

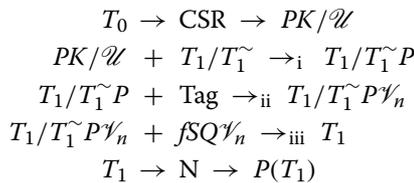
$T_1/T_1^{\sim}P\mathcal{V}_n + fSQ\mathcal{V}_n \rightarrow T_1$ which releases T_1 into the cytoplasm. The full event chain is thus:



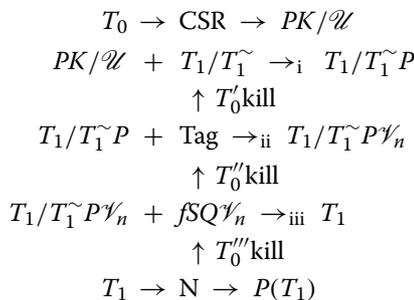
where N is the nucleus, TPT denotes the phrase *tagged protein transcription* and $P(T_1)$ indicates the protein whose construction is initiated by the trigger T_0 . Without the trigger, we see there are a variety of ways transcription can be stopped:

- T_1 does not exist in a free state; instead, it is always bound into the complex T_1/T_1^{\sim} and hence can't be activated until the T_1^{\sim} is removed.
- Any of the steps required to remove T_1^{\sim} can be blocked effectively killing transcription:
 - phosphorylation of T_1^{\sim} into $T_1^{\sim}P$ is needed so that tagging can occur. So anything that blocks the phosphorylation step will also block transcription.
 - Anything that blocks the tagging of the phosphorylated $T_1^{\sim}P$ will thus block transcription.
 - Anything that stops the removal mechanism $fSQ\mathcal{V}_n$ will also block transcription.

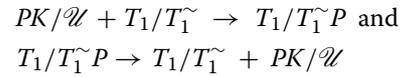
The steps above can be used therefore to further regulate the transcription of T_1 into the protein $P(T_1)$. Let T'_0 , T''_0 and T'''_0 be inhibitors of the steps above. These inhibitory proteins can themselves be regulated via triggers through mechanisms just like the ones we are discussing. In fact, $P(T_1)$ could itself serve as an inhibitory trigger - i.e. as any one of the inhibitors T'_0 , T''_0 and T'''_0 . Our theoretical pathway is now:



where the **step i**, **step ii** and **step iii** can be inhibited as shown below:



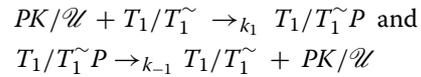
Note we have expanded to a system of four triggers which effect the outcome of $P(T_1)$. Also, note that **step i** is a phosphorylation step. Now, let's refine our analysis a bit more. Usually, reactions are paired: we typically have the competing reactions



Hence, we can imagine that **step i** is a system which is in dynamic equilibrium. The amount of $T_1/T_1^{\sim}P$ formed and destroyed forms a stable loop with no net $T_1/T_1^{\sim}P$ formed. The trigger T_0 introduces additional PK/\mathcal{U} into this stable loop and thereby effects the net production of $T_1/T_1^{\sim}P$. Thus, a new trigger T'_0 could profoundly effect phosphorylation of T_1^{\sim} and hence production of $P(T_1)$. We can see from the above comments that very fine control of $P(T_1)$ production can be achieved if we think of each step as a dynamical system in flux equilibrium. Note our discussion above is a first step towards thinking of this mechanism in terms of interacting objects.

Dynamical loop details

Our dynamical loop consists of the coupled reactions



where k_1 and k_{-1} are the forward and backward reaction rate constants and we assume the amount of T_1/T_1^{\sim} inside the cell is constant and maintained at the equilibrium concentration $[T_1/T_1^{\sim}]_e$. Since one unit of PK/\mathcal{U} combines with one unit of T_1/T_1^{\sim} to phosphorylate T_1/T_1^{\sim} to $T_1/T_1^{\sim}P$, we see

$$[T_1/T_1^{\sim}] (t) = [T_1/T_1^{\sim}]_e - [PK/\mathcal{U}] (t). \quad (1)$$

Hence, $\frac{d[PK/\mathcal{U}]}{dt} = -k_1 [PK/\mathcal{U}] [T_1/T_1^{\sim}] + k_{-1} [T_1/T_1^{\sim}P]$. For this reaction, we have $[PK/\mathcal{U}] = [T_1/T_1^{\sim}]$; thus, we find

$$\frac{d[PK/\mathcal{U}]}{dt} = -k_1 [PK/\mathcal{U}]^2 + k_{-1} [T_1/T_1^{\sim}P] \quad (2)$$

From equation 1, we see $\frac{d[T_1/T_1^{\sim}]}{dt} = -\frac{d[PK/\mathcal{U}]}{dt}$ and hence

$$\frac{d[T_1/T_1^{\sim}]}{dt} = k_1 [PK/\mathcal{U}]^2 - k_{-1} [T_1/T_1^{\sim}P] \quad (3)$$

Hence, at equilibrium, $\frac{d[T_1/T_1^{\sim}]}{dt} = 0$ implying

$$\frac{k_1}{k_{-1}} [PK/\mathcal{U}]^2 = [T_1/T_1^{\sim}P] \quad (4)$$

Now let $[PK/\mathcal{U}]_e$ denote the equilibrium concentration established by equation 4. Then if the trigger T_0 increases $[PK/\mathcal{U}]$ by $\delta_{PK/\mathcal{U}}$, we see

$$[T_1/T_1^{\sim}P]_{\text{new}} = \frac{k_1}{k_{-1}} ([PK/\mathcal{U}]_e + \delta_{PK/\mathcal{U}})^2 \quad (5)$$

which implies the percentage increase from equilibrium is $100 \left(1 + \frac{\delta_{PK/\mathcal{U}}}{[PK/\mathcal{U}]_e} \right)^2$. This *back of the envelope* calculation can be done not only at **step i** but at the other steps as well. Letting $\delta_{T_1/T_1^{\sim}P}$ and $\delta_{T_1/T_1^{\sim}P\mathcal{V}_n}$ denote changes in critical molecular concentrations, let's examine the **stage ii** equilibrium loop. We have

$$\begin{aligned} T_1/T_1^{\sim}P &\rightarrow_{k_2} T_1/T_1^{\sim}P\mathcal{V}_n \text{ and} \\ T_1/T_1^{\sim}P\mathcal{V}_n &\rightarrow_{k_{-2}} T_1/T_1^{\sim}P \end{aligned} \quad (6)$$

The kinetic equations are then

$$\frac{d[T_1/T_1^{\sim}P]}{dt} = -k_2 [T_1/T_1^{\sim}P] + k_{-2} [T_1/T_1^{\sim}P\mathcal{V}_n] \quad (7)$$

$$\frac{d[T_1/T_1^{\sim}P\mathcal{V}_n]}{dt} = k_2 [T_1/T_1^{\sim}P] - k_{-2} [T_1/T_1^{\sim}P\mathcal{V}_n] \quad (8)$$

Dynamic equilibrium then implies that $\frac{d[T_1/T_1^{\sim}P]}{dt} = \frac{d[T_1/T_1^{\sim}P\mathcal{V}_n]}{dt}$ and hence $2k_2 [T_1/T_1^{\sim}P] = 2k_{-2} [T_1/T_1^{\sim}P\mathcal{V}_n]$ or

$$\frac{k_2}{k_{-2}} [T_1/T_1^{\sim}P] = [T_1/T_1^{\sim}P\mathcal{V}_n] \quad (9)$$

Equation 9 defines the equilibrium concentrations of $[T_1/T_1^{\sim}P]_e$ and $[T_1/T_1^{\sim}P\mathcal{V}_n]_e$. Now if $[T_1/T_1^{\sim}P]$ increased to $[T_1/T_1^{\sim}P] + \delta_{T_1/T_1^{\sim}P}$, the percentage increase would be $100 \left(1 + \frac{\delta_{T_1/T_1^{\sim}P}}{[T_1/T_1^{\sim}P]_e} \right)^2$. If the increase in $[T_1/T_1^{\sim}P]$ is due to **step i**, we know

$$\begin{aligned} \delta_{T_1/T_1^{\sim}P} &= [T_1/T_1^{\sim}P]_{\text{new}} - [T_1/T_1^{\sim}P]_e \\ &= \frac{k_1}{k_{-1}} ([PK/\mathcal{U}]_e + \delta_{PK/\mathcal{U}})^2 - \frac{k_1}{k_{-1}} [PK/\mathcal{U}]_e^2 \\ &= \frac{k_1}{k_{-1}} \left\{ 2 [PK/\mathcal{U}]_e \delta_{PK/\mathcal{U}} + \delta_{PK/\mathcal{U}}^2 \right\} \end{aligned}$$

We also know $[T_1/T_1^{\sim}P]_e = \frac{k_1}{k_{-1}} [PK/\mathcal{U}]_e^2$ and therefore

$$\frac{\delta_{T_1/T_1^{\sim}P}}{[T_1/T_1^{\sim}P]_e} = 2 \frac{\delta_{PK/\mathcal{U}}}{[PK/\mathcal{U}]_e} + \left(\frac{\delta_{PK/\mathcal{U}}}{[PK/\mathcal{U}]_e} \right)^2$$

For convenience, let's define the relative change in a variable x as $r_x = \frac{\delta_x}{x}$. Thus, we can write

$$r_{T_1/T_1^{\sim}P} = \frac{\delta_{T_1/T_1^{\sim}P}}{[T_1/T_1^{\sim}P]_e} \text{ and } r_{PK/\mathcal{U}} = \frac{\delta_{PK/\mathcal{U}}}{[PK/\mathcal{U}]_e}$$

which allows us to recast the change in $[T_1/T_1^{\sim}]$ equation as

$$r_{T_1/T_1^{\sim}P} = 2r_{PK/\mathcal{U}} + r_{PK/\mathcal{U}}^2$$

Hence, it follows that $\delta_{T_1/T_1^{\sim}P\mathcal{V}_n} = \frac{k_2}{k_{-2}} \delta_{T_1/T_1^{\sim}P}$ and so

$$r_{T_1/T_1^{\sim}P\mathcal{V}_n} = r_{T_1/T_1^{\sim}P} = 2r_{PK/\mathcal{U}} + r_{PK/\mathcal{U}}^2.$$

From this, we see that trigger events which cause $r_{PK/\mathcal{U}}$ to exceed one, create an explosive increase in $[T_1/T_1^{\sim}P\mathcal{V}_n]$.

Finally, in the third step, we have

$$T_1/T_1^{\sim}P\mathcal{V}_n \rightarrow_{k_3} T_1 \text{ and } T_1 \rightarrow_{k_{-3}} T_1/T_1^{\sim}P\mathcal{V}_n \quad (10)$$

This dynamical loop can be analyzed just as we did in **step ii**. We see

$$\frac{k_3}{k_{-3}} [T_1/T_1^{\sim}P\mathcal{V}_n]_e = [T_1]_e$$

and the triggered increase in $[PK/\mathcal{U}]_e$ by $\delta_{PK/\mathcal{U}}$ induces the relative change

$$\begin{aligned} \delta_{T_1} &= \frac{k_3}{k_{-3}} \delta_{T_1/T_1^{\sim}P\mathcal{V}_n} = \frac{k_3}{k_{-3}} \frac{k_2}{k_{-2}} \delta_{T_1/T_1^{\sim}P} \\ &= \frac{k_3}{k_{-3}} \frac{k_2}{k_{-2}} \frac{k_1}{k_{-1}} \left(2 \delta_{PK/\mathcal{U}} [PK/\mathcal{U}]_e + \delta_{PK/\mathcal{U}}^2 \right) \\ &= \frac{k_3 k_2 k_1}{k_{-3} k_{-2} k_{-1}} \left((2 r_{PK/\mathcal{U}} + r_{PK/\mathcal{U}}^2) [PK/\mathcal{U}]_e^2 \right) \end{aligned}$$

We can therefore clearly see the multiplier effects of trigger T_0 on protein production T_1 which, of course, also determines changes in the production of $P(T_1)$.

The mechanism by which the trigger T_0 creates activated kinase PK/\mathcal{U} can be complex; in general, each unit of T_0 creates λ units of PK/\mathcal{U} where λ is quite large – perhaps 10,000 or more times the base level of $[PK/\mathcal{U}]_e$. Hence, if $r_{PK/\mathcal{U}} = \beta$ and $\mathcal{K} = \frac{k_1 k_2 k_3}{k_{-1} k_{-2} k_{-3}}$, we have

$$\delta_{T_1} = (2\beta + \beta^2) \mathcal{K} [PK/\mathcal{U}]_e^2 \gg [PK/\mathcal{U}]_e$$

for $\beta \gg 1$. From this quick analysis, we can clearly see the potentially explosive effect changes in T_0 can have on PK/\mathcal{U} . Let us note two very important points now: there is **richness** to this pathway and the target $P(T_1)$ can **alter hardware** or **software** easily. If $P(T_1)$ was a K^+ voltage activated gate, then we see an increase of δ_{T_1} (assuming 1 – 1 conversion of T_1 to $P(T_1)$) in the concentration of K^+ gates. This corresponds to a change in the characteristics of the axonal pulse. Similarly, $P(T_1)$ could create Na^+ gates thereby creating change in axonal pulse characteristics. $P(T_1)$ could also create other proteins whose impact on the axonal pulse is through indirect means such as the kill T'_0 etc. pathways. There is also the positive feedback pathway via $\delta_{PK/\mathcal{U}}$ through the T_0 receptor creation. Note that all of these pathways are essentially modeled by this primitive.

Second messengers

The monoamine neurotransmitters dopamine, serotonin and norepinephrine modulate the use of Ca^{+2} within a cell in a variety of ways. From the discussions in (Hille 1992), the protein levels of AC, adenylyl cyclase; CAMP, cyclic adenosine monophosphate; G proteins G_i , G_s , G_o and G_p ; PDE, phosphodiesterase; PKA, CAMP dependent kinase; CamK, calmodulin dependent kinase; PLC, phospholipase C; PKC, C kinase; IP3, inositol-1,4,5 triphosphate; DAG, diacylglycerol; IN1, PKA inhibitor; PP,

generic protein phosphatase; S_1, S_2, S_3, S_4, S_6 and S_7 , surface proteins, and others, can all be altered. Creation of these proteins alters the hardware of the cell; indeed, these proteins S_i could be anything used in the functioning of the cell. Following (Bray 1998), we can infer protein kinases, protein phosphatases, transmembrane receptors etc. that carry and process messages inside the cell are associated with compact clusters of molecules attached to the cell membrane which *operate as functional units and signaling complexes provide an intermediate level of organization like the integrated circuits used in VLSI. However, this interpretation is clearly flawed as these circuits are self-modifying.* A close look extracellular triggers abstractly helps us understand how to approximate their effects. The full details showing how we obtain abstractions of information processing can be found in (Peterson 2014b) and here, we will be brief. Let T_0 denote a second messenger trigger which moves through a port P to create a new trigger T_1 some of which binds to B_1 . We have already discussed this is an abstract way. Here we are looking at it more computationally. A schematic of this is shown in Figure 1. In the figure, r is a number between 0 and 1 which represents the fraction of the trigger T_1 which is free in the cytosol. Hence, $100r\%$ of T_1 is free and $100(1 - r)$ is bound to B_1 creating a storage complex B_1/T_1 . For our simple model, we assume rT_1 is transported to the nuclear membrane where some of it binds to the enzyme E_1 . Let s in $(0, 1)$ denote the fraction of rT_1 that binds to E_1 . We illustrate this in Figure 2.

We denote the complex formed by the binding of E_1 and T_1 by E_1/T_1 . From Figure 2, we see that the proportion of T_1 that binds to the genome (DNA) and initiates protein creation $P(T_1)$ is thus srT_1 .

The protein created, $P(T_1)$, could be many things. Here, let us assume that $P(T_1)$ is a sodium, Na^+ , gate. Thus, our high level model is $sE_1/rT_1 + DNA \rightarrow Na^+$ gate. We therefore increase the concentration of Na^+ gates, $[Na^+]$ thereby creating an increases in the sodium conductance, g_{Na} . The standard Hodgkin - Huxley conductance model (details are in (Peterson 2014c)) is given by $g_{Na}(t, V) = g_{Na}^{max} \mathcal{M}_{Na}^p(t, v) \mathcal{H}_{Na}^q(t, V)$ where t is time and V is membrane voltage. The variables \mathcal{M}_{Na} and \mathcal{H}_{Na} are the activation and inactivation functions for the sodium gate with p and q appropriate positive powers. Finally,

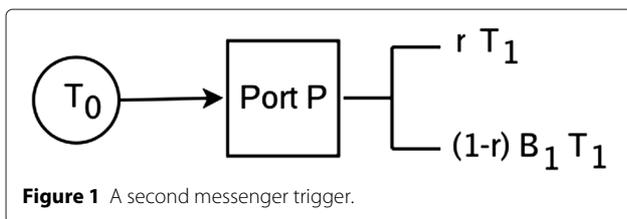


Figure 1 A second messenger trigger.

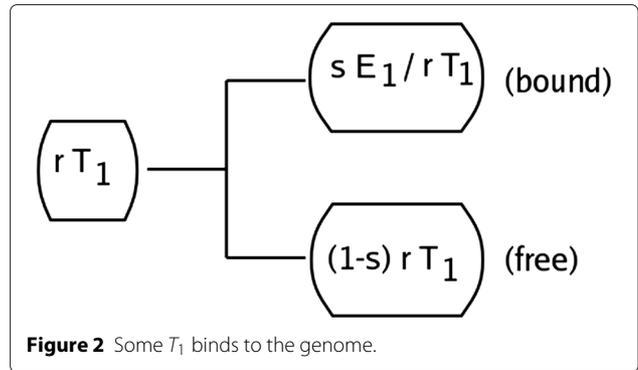


Figure 2 Some T_1 binds to the genome.

g_{Na}^{max} is the maximum conductance possible. These models generate \mathcal{M}_{Na} and \mathcal{H}_{Na} values in the range $(0, 1)$ and hence, $0 \leq g_{Na}(t, V) \leq g_{Na}^{max}$.

We can model increases in sodium conductances as increases in g_{Na}^{max} with efficiency e , where e is a number between 0 and 1. We will not assume all of the $sE_1/rT_1 + DNA$ to sodium gate reaction is completed. It follows that e is similar to a Michaelson - Mentin kinetics constant. We could also alter activation, \mathcal{M}_{Na} , and/or inactivation, \mathcal{H}_{Na} , as functions of voltage, V in addition to the change in the maximum conductance. However, we are interested in a simple model at present. Our full schematic is then given in Figure 3.

We can model the choice process, rT_1 or $(1 - r)B_1/T_1$ via a simple sigmoid, $f(x) = 0.5 \left(1 + \tanh \left(\frac{x-x_0}{g} \right) \right)$ where the transition rate at x_0 is $f'(x_0) = \frac{1}{2g}$. Hence, the "gain" of the transition can be adjusted by changing the value of g . We assume g is positive. This function can be interpreted as switching from of "low" state 0 to a high state 1 at speed $\frac{1}{2g}$. Now the function $h = rf$ provides an output in (r, ∞) . If x is larger than the threshold x_0 , h rapidly transitions to a high state r . On the other hand, if x is below threshold, the output remains near the low state 0 .

We assume the trigger T_0 does not activate the port P unless its concentrations is past some threshold $[T_0]_b$ where $[T_0]_b$ denotes the *base* concentration. Hence, we can model the port activity by $h_p([T_0]) = \frac{r}{2} \left(1 + \tanh \left(\frac{[T_0]-[T_0]_b}{g_p} \right) \right)$ where the two shaping parameters g_p (transition rate) and $[T_0]_b$ (threshold) must be chosen. We can thus model the schematic of Figure 1 as $h_p([T_0]) [T_1]_n$ where $[T_1]_n$ is the nominal concentration of the induced trigger T_1 . In a similar way, we let $h_e(x) = \frac{s}{2} \left(1 + \tanh \left(\frac{x-x_0}{g_e} \right) \right)$ Thus, for $x = h_p([T_0]) [T_1]_n$, we have h_e is a switch from 0 to s . Note that $0 \leq x \leq r[T_1]_n$ and so if $h_p([T_0]) [T_1]_n$ is close to $r[T_1]_n$, h_e is approximately s . Further, if $h_p([T_0]) [T_1]_n$ is small, we will have h_e is close to 0. This suggests a threshold value for h_e of $\frac{r[T_1]_n}{2}$. We conclude

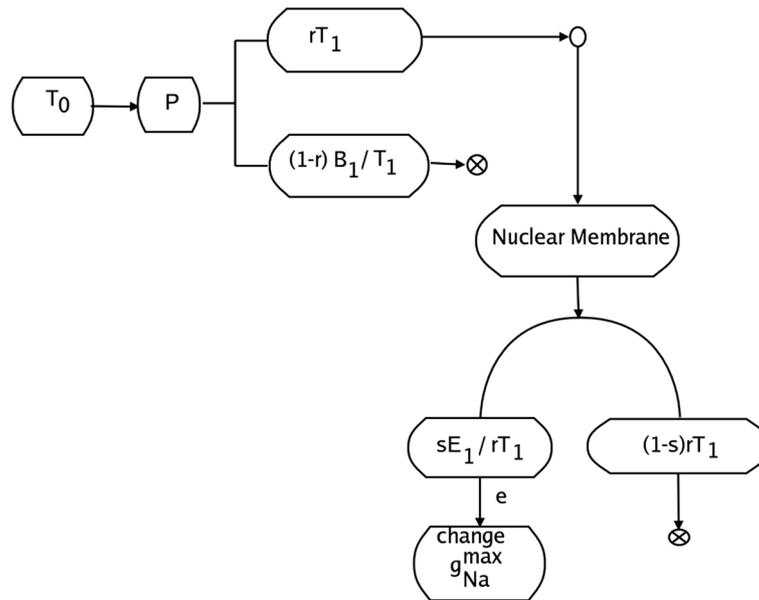


Figure 3 Maximum sodium conductance control pathway.

$$h_e(h_p([T_0]) [T_1]_n) = \frac{s}{2} \left(1 + \tanh \left(\frac{h_p([T_0]) [T_1]_n - \frac{r[T_1]_n}{2}}{g_e} \right) \right)$$

which lies in $[0, s)$. This is the amount of activated T_1 which reaches the genome to create the target protein $P(T_1)$. It follows then that $[P(T_1)] = h_e(h_p([T_0]) [T_1]_n) [T_1]_n$. The protein is created with efficiency e and so we model the conversion of $[P(T_1)]$ into a change in g_{Na}^{max} as follows. Let

$$h_{Na}(x) = \frac{e}{2} \left(1 + \tanh \left(\frac{x - x_0}{g_{Na}} \right) \right)$$

which has output in $[0, e)$. Here, we want to limit how large a change we can achieve in g_{Na}^{max} . Hence, we assume there is an upper limit which is given by $\Delta g_{Na}^{max} = \delta_{Na} g_{Na}^{max}$. Thus, we limit the change in the maximum sodium conductance to some percentage of its baseline value. It follows that $h_{Na}(x)$ is about δ_{Na} if x is sufficiently large and small otherwise. This suggests that x should be $[P(T_1)]$ and since translation to $P(T_1)$ occurs no matter how low $[T_1]$ is, we can use a switch point value of $x_0 = 0$. We conclude

$$h_{Na}([P(T_1)]) = \frac{e}{2} \delta_{Na} g_{Na}^{max} \left(1 + \tanh \left(\frac{[P(T_1)]}{g_{Na}} \right) \right) \quad (11)$$

Our model of the change in maximum sodium conductance is therefore $\Delta g_{Na}^{max} = h_{Na}([P(T_1)])$. We can thus

alter the action potential via a second messenger trigger by allowing

$$g_{Na}(t, V) = (g_{Na}^{max} + h_{Na}([P(T_1)])) \times \mathcal{M}_{Na}^p(t, V) \mathcal{H}_{Na}^q(t, V)$$

for appropriate values of p and q within a standard Hodgkin - Huxley model.

Next, if we assume a modulatory agent acts as a trigger T_0 as described above, we can generate action potential pulses using the standard Hodgkin - Huxley model for a large variety of critical sodium trigger shaping parameters. We label these with a Na to indicate their dependence on the sodium second messenger trigger. $[r^{Na}, [T_0]_b^{Na}, g_p^{Na}, s^{Na}, g_e^{Na}, e^{Na}, g_{Na}, \delta_{Na}]'$ We can follow the procedure outlined in this section for a variety of triggers. We therefore can add a potassium gate trigger with shaping parameters $[r^K, [T_0]_b^K, g_p^K, s^K, g_e^K, e^K, g_K, \delta_K]'$.

Concatenated sigmoid transitions:

In the previous section, we have found how to handle alterations in g_{Na}^{max} due to a trigger T_0 . We have

$$g_{Na}(t, V) = (g_{Na}^{max} + h_{Na}([P(T_1)])) \mathcal{M}_{Na}^p(t, V) \mathcal{H}_{Na}^q(t, V)$$

where

$$h_{Na}([P(T_1)]) = \frac{e}{2} \delta_{Na} g_{Na}^{max} \left(1 + \tanh \left(\frac{[P(T_1)]}{g_{Na}} \right) \right)$$

with e and δ_{Na} in $(0, 1)$. Let $\sigma(x, x_0, g_0) = \frac{1}{2} \left(1 + \tanh\left(\frac{x-x_0}{g_0}\right) \right)$. Then the sodium conductance modification equation is $h_{Na}([P(T_1)]) = e\delta_{Na}g_{Na}^{max}\sigma([P(T_1)], 0, g_{Na})$. Using this same notation, we see

$$\begin{aligned} h_p([T-0]) &= r\sigma([T_0], [T_0]_b, g_p) \\ h_e(h_p([T_0])[T_1]_n) &= s\sigma\left(h_p([T_0])[T_1]_n, \frac{r[T_1]_n}{2}, g_e\right) \\ &= s\sigma\left(r\sigma([T_0], [T_0]_b, g_p)[T_1]_n, \frac{r[T_1]_n}{2}, g_e\right) \end{aligned}$$

Note the concatenation of the sigmoidal processing. Now $[P(T_1)] = h_e(h_p([T_0])[T_1]_n)[T_1]_n$. Thus,

$$[P(T_1)] = s\sigma\left(r\sigma([T_0], [T_0]_b, g_p)[T_1]_n, \frac{r[T_1]_n}{2}, g_e\right)[T_1]_n.$$

Finally,

$$g_{Na}(t, V) = g_{Na}^{max} (1 + e\delta_{Na} \sigma([P(T_1)], 0, g_{Na})) \mathcal{M}_{Na}^p(t, V) \mathcal{H}_{Na}^q(t, V)$$

Implicit in this formula is the *cascade* “ $\sigma(\sigma(\sigma))$ ” as $\sigma([P(T_1)], 0, g_{Na})$ uses two concatenated sigmoid calculations itself. We label this as a σ_3 transition and use the notation

- $\sigma_3([T_0], [T_0]_b, g_p$; inner most sigmoid
- r ; scale innermost calculation by r
- $[T_1]_n$; scale again by $[T_1]_n$; this is input to next sigmoid
- $\frac{r[T_1]_n}{2}, g_e$; offset and gain of next sigmoid
- s ; scale results by s
- $[T_1]_n$; scale again by $[T_1]_n$; this is $[P(T_1)]$, input into last sigmoid
- $0, g_{Na}$; offset and gain of last sigmoid)

Thus, the g_{Na} computation can be written as

$$\begin{aligned} g_{Na}(t, V) &= g_{Na}^{max} \left(1 + e\delta_{Na} \sigma_3 \left([T_0], [T_0]_b, g_p; r; [T_1]_n; \frac{r[T_1]_n}{2}, g_e; s; [T_1]_n; 0, g_{Na} \right) \right) \\ &\times \mathcal{M}_{Na}^p(t, V) \mathcal{H}_{Na}^q(t, V) \end{aligned}$$

This implies a trigger T_0 has associated with it a data vector

$$W_{T_0} = \left[[T_0], [T_0]_b, g_p, r, [T_1]_n, \frac{r[T_1]_n}{2}, g_e, s, [T_1]_n, 0, g_T \right]'$$

where g_T denotes the final gain associated with the third level sigmoidal transition to create the final gate product. We can then rewrite our modulation equation as

$$g_{Na}(t, V) = g_{Na}^{max} (1 + e\delta_{Na} \sigma_3(W_{Na})) \mathcal{M}_{Na}^p(t, V) \mathcal{H}_{Na}^q(t, V)$$

A graphic computation model

We can also recast this model into computational graph structure. This makes it easier to see how the calculations will be performed as asynchronous agents. Consider the typical standard sigmoid transformation $h_p([T_0]) = \frac{r}{2} \left(1 + \tanh\left(\frac{[T_0]-[T_0]_b}{g_p}\right) \right)$. We can draw this as a graph as is shown in Figure 4 where h denotes the standard sigmoidal state transition function.

We also have $h_e(h_p([T_0])[T_1]_n) = \frac{s}{2} \left(1 + \tanh\left(\frac{h_p([T_0])[T_1]_n - \frac{r[T_1]_n}{2}}{g_e}\right) \right)$ which for a given output Y becomes

$$h_e(Y) = \frac{s}{2} \left(1 + \tanh\left(\frac{Y - \frac{r[T_1]_n}{2}}{g_e}\right) \right)$$

and the computation can be represented graphically by Figure 5. Finally, we have used $[P(T_1)] = h_e(h_p([T_0])[T_1]_n)[T_1]_n$ and $h_{Na}([P(T_1)]) = e\delta_{Na}g_{Na}^{max}h([P(T_1)], 0, g_{Na})$ which can be shown diagrammatically as in Figure 6.

These graphs are, of course, becoming increasingly complicated. However, they are quite useful in depicting how feedback pathways can be added to our computations. Let's add feedback pathways to the original maximum conductance control pathway we illustrated in Figure 3. Figure 7 is equivalent to that of the computational graph of Figure 6 but shows more of the underlying cellular mechanisms.

The feedback pathways are indicated by the variables ξ_0 and ξ_1 . The feedback ξ_0 is meant to suggest that some of the protein kinase T_1 which is bound onto $(1-r)B_1/T_1$ is recycled (probably due to other governing cycles) back to *free* T_1 . We show this with a line drawn back to the port P . Similarly, some of the protein kinase not used for binding

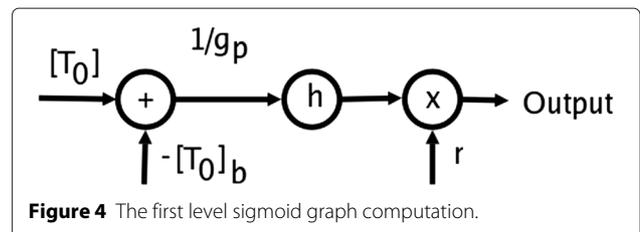


Figure 4 The first level sigmoid graph computation.

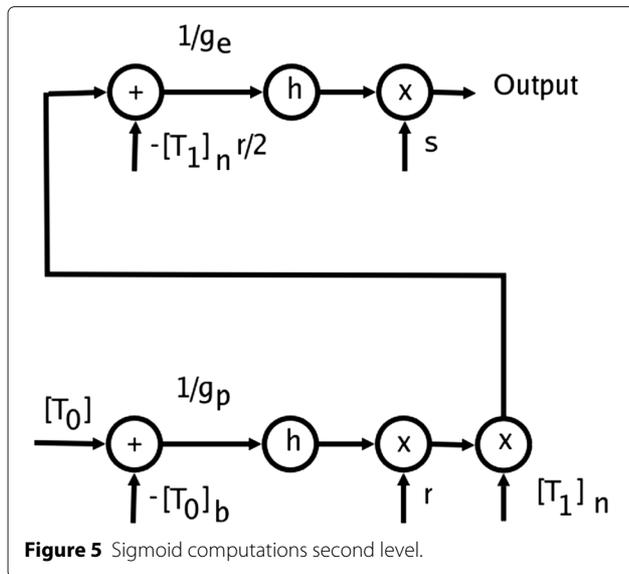


Figure 5 Sigmoid computations second level.

to the enzyme E_1 , to start the protein creation process culminating in $P(T_1)$, is allowed to be freed for reuse. This is shown with the line labeled with the legend $1 - \xi_1$ leading back to the nuclear membrane. It is clear there are many other feedback possibilities. Also the identity of the protein $P(T_1)$ is fluid. We have already discussed the cases where $P(T_1)$ can be voltage dependent gates for sodium

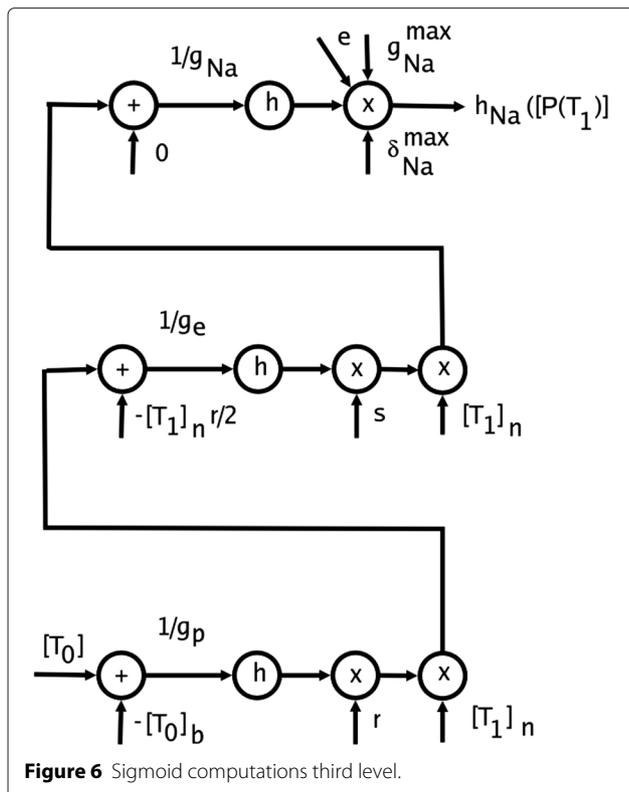


Figure 6 Sigmoid computations third level.

and potassium and calcium second messenger proteins. However, there are other possibilities:

- B_1 : thereby increasing T_1 binding
- T_1 : thereby increasing rT_1 and $P(T_1)$
- E_1 thereby increasing $P(T_1)$.

and so forth. We can also specialize this to the case of Ca^{++} triggers, but we will not do so here.

Let's specialize our discussion to the case of a neurotransmitter trigger. When two cells interact via a synaptic interface, the electrical signal in the pre-synaptic cell triggers a release of a neurotransmitter (NT) from the pre-synapse which crosses the synaptic cleft and then by docking to a port on the post cell, initiates a post-synaptic cellular response. The general pre-synaptic mechanism consists of several key elements: one, NT synthesis machinery so the NT can be made locally; two, receptors for NT uptake and regulation; three, enzymes that package the NT into vesicles in the pre-synapse membrane for delivery to the cleft. There are two general pre-synaptic types: *monoamine* and *peptide*. In the *monoamine* case, all three elements for the pre-cell response are first manufactured in the pre-cell using instructions contained in the pre-cell's genome and shipped to the pre-synapse. Hence, the monoamine pre-synapse does not require further instructions from the pre-cell genome and response is therefore fast. The *peptide* pre-synapse can only manufacture a *peptide* neurotransmitter in the pre-cell genome; if a peptide neurotransmitter is needed, there is a lag in response time. Also, in the peptide case, there is no re-uptake pump so peptide NT can't be reused.

On the post-synaptic side, the fast response is triggered when the bound NT/ Receptor complex initiates an immediate change in ion flux through the gate thereby altering the electrical response of the post cell membrane and hence, ultimately its action potential and spike train pattern. Examples are glutamate (excitatory) and GABA (inhibitory) neurotransmitters. The slow response occurs when the initiating NT triggers a second messenger response in the interior of the cell. When the output of one neuron is sent into the input system of another, we typically call the neuron providing the input the *pre-neuron* and the neuron generating the output, the *post-neuron*. The post-neuron's output is a classical action potential which we will eventually approximate using a low dimensional vector called the Biological Feature Vector or BFV. The pre-neuron generates an axonal pulse which influences the release of the contents of synaptic vesicles into the fluid contained in the region between the neurons. The vesicles contain *neurotransmitters*. For convenience, we focus on one such neurotransmitter, labeled ζ . The ζ neurotransmitter then acts like the trigger T_0 we have already discussed. It binds in some fashion to a

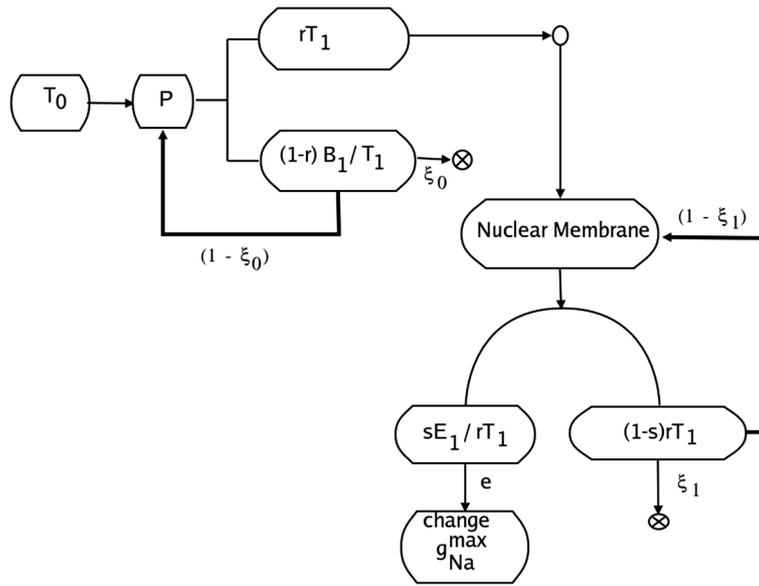


Figure 7 Adding feedback maximum sodium conductance control pathway.

port or gate specialized for the ζ neurotransmitter. The neurotransmitter ζ then initiates a cascade of reactions:

- It passes through the gate, entering the interior of the dendrite. It then forms a complex, $\hat{\zeta}$.
- Inside the post-dendrite, $\hat{\zeta}$ influences the passage of ions through the cable wall. For example, it may increase the passage of Na^+ through the membrane of the cable thereby initiating an ESP. It could also influence the formation of a calcium current, an increase in K^+ and so forth.
- The influence via $\hat{\zeta}$ can be that of a second messenger trigger.

Each neuron creates a brew of neurotransmitters specific to its type. A trigger of type T_0 can thus influence the production of neurotransmitters with concomitant changes in post-neuron activity.

Methods

The abstract neuron model

It is clear neuron classes can have different trigger characteristics. First, consider the case of neurons which create the monoamine neurotransmitters. Neurons of this type in the Reticular Formation of the midbrain produce a monoamine neurotransmitter packet at the synaptic junction between the axon of the pre-neuron and the dendrite of the post-neuron. The monoamine neurotransmitter is then released into the synaptic cleft and it induces a second messenger response. The strength of this response is dependent on the BFV input from pre-neurons that form synaptic connections with the post-neuron. The strength of this input determines the strength of the monoamine

trigger into the post-neuron dendrite. Let the strength for neurotransmitter ζ be given by the weighting term $c_{pre,post}^\zeta$. The trigger at time t and dendrite location w on the dendritic cable is therefore

$$T_0(t, w) = \frac{c_{pre,post}^\zeta}{\sqrt{t - t_0}} \exp\left(\frac{-(w - w_0)^2}{4D_0^\zeta(t - t_0)}\right).$$

where D_0^ζ is the diffusion constant associated with the trigger. The trigger T_0 has associated with it the protein T_1 . We let

$$T_1(t, w) = \frac{d_{pre,post}^\zeta}{\sqrt{t - t_0}} \exp\left(\frac{-(w - w_0)^2}{4D_1^\zeta(t - t_0)}\right).$$

where $d_{pre,post}^\zeta$ denotes the strength of the induced T_1 response and D_1^ζ is the diffusion constant of the T_1 protein. This trigger will act through the usual pathway. Also, we let T_2 denote the protein $P(T_1)$. T_2 transcribes a protein target from the genome with efficiency e .

$$h_p([T_0(t, w)]) = \frac{r}{2} \left(1 + \tanh\left(\frac{[T_0(t, w)] - [T_0]_b}{g_p}\right) \right)$$

$$I(t, w) = \frac{h_p([T_0(t, w)]) [T_1]_n}{\sqrt{t - t_0}} \exp\left(\frac{-(w - w_0)^2}{4D_1^\lambda(t - t_0)}\right)$$

$$h_e(I(t, w)) = \frac{s}{2} \left(1 + \tanh\left(\frac{I(t, w) - \frac{r[T_1]_n}{2}}{g_e}\right) \right)$$

$$[P(T_1)](t, w) = h_e(I(t, w))$$

$$h_{T_2}(t, w) = \frac{e}{2} \left(1 + \tanh\left(\frac{[T_2](t, w)}{g_{T_2}}\right) \right)$$

$$[T_2](t, w) = h_{T_2}(t, w)[T_2]_n$$

Note $[T_2](t, w)$ gives the value of the protein T_2 concentration at some discrete time t and spatial location w . This response can also be modulated by feedback. In this case, let ξ denote the feedback level. Then, the final response is altered to $h_{T_2}^f$ where the superscript f denotes the feedback response and the constant ω is the strength of the feedback; we have $h_{T_2}^f(t, w) = \omega \frac{1}{\xi} h_{T_2}(t, w)$ and $[T_2](t, w) = h_{T_2}^f(t, w)[T_2]_n$. There are a large number of shaping parameters here. For example, for each neurotransmitter, we could alter the parameters due to calcium trigger diffusion. These would include D_0^ξ , the diffusion constant for the trigger, and D_1^ξ , the diffusion constant for the gate induced protein T_1 . In addition, transcribed proteins could alter – we know their first order quantitative effects due to our earlier analysis – $d_{pre,post}^\xi$, the strength of the T_1 response, r , the fraction of T_1 free, g_p , the trigger gain, $[T_0]_b$, the trigger threshold concentration, s , the fraction of active T_1 reaching genome, g_e , the trigger gain for active T_1 transition, $[T_1]_n$, the threshold for T_1 , $[T_2]_n$, the threshold for $P(T_1) = T_2$, g_{T_2} , the gain for T_2 , ω , the feedback strength, and ξ , the feedback amount for $T_1 = 1 - \xi$. Note $d_{pre,post}^\xi$ could be simply $c_{pre,post}^\lambda$. The neurotransmitter triggers can alter many parameters important to the creation of the BFV. For example, the maximum sodium and potassium conductances can be altered via the equation for T_2 . For sodium, $T_2(t, w) = h_{T_2}(t, w)[T_2]_n$ becomes

$$\begin{aligned}
 [T_2]_n &= \delta_{Na} g_{Na}^{max} \\
 h_{T_2}^f(t, w) &= \omega \frac{1}{\xi} h_{T_2}(t, w) \\
 [T_2](t, w) &= h_{T_2}^f(t, w)[T_2]_n \\
 g_{Na}(t, w, V) &= (g_{Na}^{max} + [T_2](t, w)) \mathcal{M}_{Na}^p(t, V) \mathcal{H}_{Na}^q(t, V)
 \end{aligned}$$

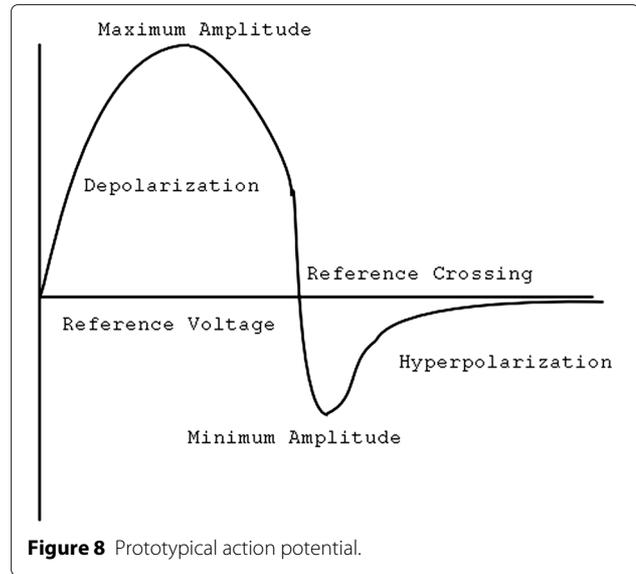
There would be a similar set of equations for potassium. Finally, neurotransmitters and other second messenger triggers have delayed effects in general. So if the trigger T_0 binds with a port P at time t_0 , the changes in protein levels $P(T_1)$ might also need to be delayed by a factor τ^ξ .

Abstract neuron design

We can see the general structure of a typical action potential is illustrated in Figure 8.

We can use the following points on this generic action potential to construct a low dimensional feature vector of Equation 12.

$$\xi = \begin{cases} (t_0, V_0) & \text{start point} \\ (t_1, V_1) & \text{maximum point} \\ (t_2, V_2) & \text{return to reference voltage} \\ (t_3, V_3) & \text{minimum point} \\ (g, t_4, V_4) & \text{sigmoid tail model} \end{cases} \quad (12)$$



where the model of the tail of the action potential is of the form $V_m(t) = V_3 + (V_4 - V_3) \tanh(g(t - t_3))$. Note that $V'_m(t_3) = (V_4 - V_3) g$ and so if we were using real voltage data, we would approximate $V'_m(t_3)$ by a standard finite difference. The biological feature vector therefore stores many of the important features of the action potential in a low dimensional form. We note these include

- The interval $[t_0, t_1]$ is the duration of the rise phase. This interval can be altered or modulated by neurotransmitter activity on the nerve cell's membrane as well as second messenger signaling from within the cell.
- The height of the pulse, V_1 , is an important indicator of excitation.
- The time interval between the highest activation level, V_1 and the lowest, V_3 , is closely related to spiking interval. This time interval, $[t_1, t_3]$, is also amenable to alteration via neurotransmitter input.
- The *height* of the depolarizing pulse, V_4 , helps determine how long it takes for the neuron to reestablish its reference voltage, V_0 .
- The neuron voltage takes time to reach reference voltage after a spike. This is the time interval by the interval $[t_3, \infty]$.
- The exponential rate of increase in the time interval $[t_3, \infty]$ is also very important to the regaining of nominal neuron electrophysiological characteristics.

We have shown the BFV captures the characteristics of the output pulse well enough to classify neurotransmitter inputs on the basis of how they change the BFV (Peterson and Khan 2006) and we will now use modulations of the BFV induced by second messengers in our nodal computations. The feature vector output of a neural object

is due to the cumulative effect of second messenger signaling to the genome of this object which influences the action potential and thus feature vector of the object by altering its complicated mixture of ligand and voltage activated ion gates, enzymes and so forth. We can then define an algebra of output interactions that we can use in building the models. We motivate our approach using the basic Hodgkin - Huxley model which depends on a large number of parameters. Of course, more sophisticated action potential models can be used, but the standard two ion gate Hodgkin - Huxley model is sufficient for our needs here.

Using the vector ξ from Equation 12, we can construct the BVF. Note for the sigmoid tail model, we have $V'_m(t_3) = (V_4 - V_3) g$ and we can approximate $V'_m(t_3)$ by a standard finite difference. We pick a data point (t_5, V_5) that occurs after the minimum – typically we use the voltage value at the time t_5 that is 5 time steps downstream from the minimum and approximate the derivative at t_3 by $V'_m(t_3) \approx \frac{V_5 - V_3}{t_5 - t_3}$. The value of g is then determined to be $g = \frac{V_5 - V_3}{(V_4 - V_3)(t_5 - t_3)}$ which reflects the asymptotic nature of the hyperpolarization phase of the potential. Clearly, we can model an inhibitory pulse, *mutatis mutandi*.

The BFV functional form

In Figure 9, we indicated the three major portions of the biological feature vector and the particular data points chosen from the action potential which are used for the

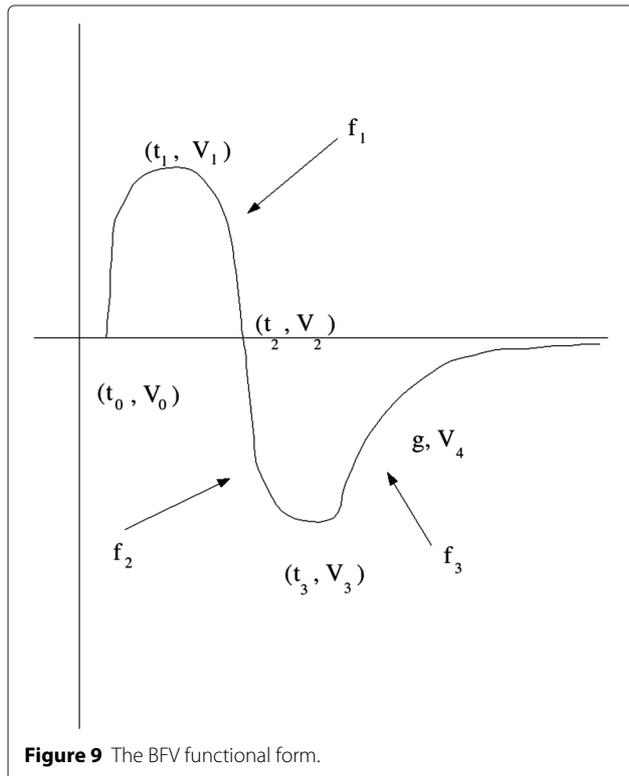


Figure 9 The BFV functional form.

model. These are the two parabolas f_1 and f_2 and the sigmoid f_3 . The parabola f_1 is treated as the two distinct pieces f_{11} and f_{12} given by

$$f_{11}(t) = a^{11} + b^{11}(t - t_1)^2 \tag{13}$$

$$f_{12}(t) = a^{12} + b^{12}(t - t_1)^2 \tag{14}$$

Thus, f_1 consists of two joined parabolas which both have a vertex at t_1 . The functional form for f_2 is a parabola with vertex at t_3 : $f_2(t) = a^2 + b^2(t - t_3)^2$. Finally, the sigmoid portion of the model is given by $f_3(t) = V_3 + (V_4 - V_3) \tanh(g(t - t_3))$. We have also simplified the BFV even further by dropping the explicit time point t_4 and modeling the portion of the action potential after the minimum voltage by the sigmoid f_3 . From the data, it follows that

$$\begin{aligned} f_{11}(t_0) &= V_0 = a^2 + b^{11}(t_0 - t_1)^2, \text{ and} \\ f_{11}(t_1) &= V_1 = a^{11} \quad f_{12}(t_1) = V_1 = a^{12}, \text{ and} \\ f_{11}(t_2) &= V_2 = a^{12} + b^{12}(t_2 - t_1)^2 \end{aligned}$$

This implies

$$\begin{aligned} a^{11} &= V_1, \quad b^{11} = \frac{V_0 - V_1}{(t_0 - t_1)^2}, \quad a^{12} = V_1, \text{ and} \\ b^{12} &= \frac{V_2 - V_1}{(t_2 - t_1)^2} \end{aligned}$$

In a similar fashion, the f_2 model is constrained by

$f_2(t_2) = V_2 = a^2 + b^2(t_2 - t_3)^2$ and $f_2(t_3) = V_3 = a^2$. We conclude that $a^2 = V_3$ and $b^2 = \frac{V_2 - V_3}{(t_2 - t_3)^2}$. Hence, the functional form of the BFV model can be given by the mapping f of equation 15.

$$f(t) = \begin{cases} V_1 + \frac{V_0 - V_1}{(t_0 - t_1)^2} (t - t_1)^2, & t_0 \leq t \leq t_1 \\ V_1 + \frac{V_2 - V_1}{(t_2 - t_1)^2} (t - t_1)^2, & t_1 \leq t \leq t_2 \\ V_3 + \frac{V_2 - V_3}{(t_2 - t_3)^2} (t - t_3)^2, & t_2 \leq t \leq t_3 \\ V_4 + (V_4 - V_3) \tanh(g(t - t_3)), & t_3 \leq t < \infty \end{cases} \tag{15}$$

All of our parabolic models can also be written in the form $p(t) = \pm \frac{1}{4\beta} (t - \alpha)^2$ where 4β is the width of the line segment through the focus of the parabola. The models f_{11} and f_{12} point down and so use the “minus” sign while f_2 uses the “plus”. By comparing our model equations with this generic parabolic equation, we find the width of the parabolas of f_{11} , f_{12} and f_2 is given by

$$\begin{aligned} 4\beta_{11} &= \frac{(t_0 - t_1)^2}{V_1 - V_0} = \frac{-1}{b^{11}}, \quad 4\beta_{12} = \frac{(t_2 - t_1)^2}{V_1 - V_2} = \frac{-1}{b^{12}} \text{ and} \\ 4\beta_2 &= \frac{(t_2 - t_3)^2}{V_2 - V_3} = \frac{1}{b^2}. \end{aligned}$$

Modulation of the BFV parameters

We want to modulate the output of our abstract neuron model by altering the BFV. The BFV itself consists of 11 parameters, but better insight, into how alterations of

the BFV introduce changes in the *action potential* we are creating, comes from studying changes in the mapping f given in Section “The BFV functional form”. In addition to changes in timing, t_0 , t_1 , t_2 and t_3 , we can also consider the variations of Equation 16.

$$\begin{bmatrix} \Delta a^{11} \\ \Delta b^{11} \\ \Delta a^{12} \\ \Delta b^{12} \\ \Delta a^2 \\ \Delta b^2 \end{bmatrix} = \begin{bmatrix} \Delta V_1 \\ \Delta \left(\frac{V_0 - V_1}{(t_0 - t_1)^2} \right) \\ \Delta V_1 \\ \Delta \left(\frac{V_2 - V_1}{(t_2 - t_1)^2} \right) \\ \Delta V_3 \\ \Delta \left(\frac{V_2 - V_3}{(t_2 - t_3)^2} \right) \end{bmatrix} \quad (16)$$

$$= \begin{bmatrix} \Delta \text{Maximum Voltage} \\ \Delta \left(\frac{-1}{4\beta_{11}} \right) \\ \Delta \text{Maximum Voltage} \\ \Delta \left(\frac{-1}{4\beta_{12}} \right) \\ \Delta \text{Minimum Voltage} \\ \Delta \left(\frac{1}{4\beta_2} \right) \end{bmatrix}$$

It is clear that modulatory inputs that alter the cap shape and hyperpolarization curve of the BFV functional form can have a profound effect on the information contained in the “action potential”. For example, a hypothetical neurotransmitter that alters V_1 will also alter the latis rectum distance across the cap f_1 . Further, direct modifications to the latis rectum distance in any of the two caps f_{11} and f_{12} can induce corresponding changes in times t_0 , t_1 and t_2 and voltages V_0 , V_1 and V_2 . A similar statement can be made for changes in the latis rectum of cap f_2 . For example, if a neurotransmitter induced a change of, say 1% in $4\beta_{11}$, this would imply that $\Delta \left(\frac{V_1 - V_0}{(t_0 - t_1)^2} \right) = .04\beta_{11}^0$ where β_{11}^0 denotes the original value of β_{11}^0 . Thus, to first order

$$\begin{aligned} .04\beta_{11}^0 &= \left(\frac{\partial \beta_{11}}{\partial V_0} \right)^* \Delta V_0 + \left(\frac{\partial \beta_{11}}{\partial V_1} \right)^* \Delta V_1 + \left(\frac{\partial \beta_{11}}{\partial t_0} \right)^* \Delta t_0 \\ &+ \left(\frac{\partial \beta_{11}}{\partial t_1} \right)^* \Delta t_1 \end{aligned} \quad (17)$$

where the superscript * on the partials indicates they are evaluated at the base point (V_0, V_1, t_0, t_1) . Taking partials we find

$$\begin{aligned} \frac{\partial \beta_{11}}{\partial V_0} &= 2 \frac{(t_0 - t_1)^2}{(V_1 - V_0)^2} = \frac{2}{V_1 - V_0} \beta_{11}^0, \\ \frac{\partial \beta_{11}}{\partial V_1} &= -2 \frac{(t_0 - t_1)^2}{(V_1 - V_0)^2} = -\frac{2}{V_1 - V_0} \beta_{11}^0, \\ \frac{\partial \beta_{11}}{\partial t_0} &= 2 \frac{t_0 - t_1}{V_1 - V_0} = \frac{2}{t_0 - t_1} \beta_{11}^0, \\ \frac{\partial \beta_{11}}{\partial t_1} &= -2 \frac{t_0 - t_1}{V_1 - V_0} = -\frac{2}{t_0 - t_1} \beta_{11}^0 \end{aligned}$$

Thus, Equation 17 becomes

$$\begin{aligned} .04\beta_{11}^0 &= \frac{2\Delta V_0}{V_1 - V_0} \beta_{11}^0 - \frac{2\Delta V_1}{V_1 - V_0} \beta_{11}^0 + 2\Delta t_0 \frac{1}{t_0 - t_1} \beta_{11}^0 \\ &- 2\Delta t_1 \frac{1}{t_0 - t_1} \beta_{11}^0 \end{aligned}$$

This simplifies to

$$\begin{aligned} .02(V_1 - V_0)(t_0 - t_1) &= (\Delta V_0 - \Delta V_1)(t_0 - t_1) \\ &- (\Delta t_0 - \Delta t_1)(V_1 - V_0) \end{aligned}$$

Since we can do this analysis for any percentage r of β_{11}^0 , we can infer that a neurotransmitter that modulates the action potential by perturbing the “width” or latis rectum of the cap of f_{11} can do so satisfying the equation

$$\begin{aligned} 2r(V_1 - V_0)(t_1 - t_0) &= (\Delta V_0 - \Delta V_1)(t_0 - t_1) \\ &- (\Delta t_0 - \Delta t_1)(V_1 - V_0) \end{aligned}$$

Similar equations can be derived for the other two width parameters for caps f_{12} and f_3 . These sorts of equations give us design principles for complex neurotransmitter modulations of a BFV.

Modulation via the BFV ball and stick model

The BFV model we build consists of a dendritic system and a computational core which processes BFV input sequence to generate a BFV output. The standard Hodgkin-Huxley equations tell us

$$\begin{aligned} C_m \frac{dV_M}{dt} &= I_E - g_K^{Max}(\mathcal{M}_K)^4(V_m, t)(V_m - E_K) \\ &- g_{Na}^{Max}(\mathcal{M}_{NA})^3(V_m, t)(\mathcal{H}_{NA})(V_m, t) \\ &\times (V_m - E_{Na}) - g_L(V_m - E_L). \end{aligned}$$

Since the BFV is structured so that the action potential has a maximum at t_1 of value V_1 and a minimum at t_3 of value V_3 , we have $V'_m(t_1) = 0$ and $V'_m(t_3) = 0$. This gives

$$\begin{aligned} I_E(t_1) &= g_K^{Max}(\mathcal{M}_K)^4(V_1, t_1)(V_1 - E_K) \\ &+ g_{Na}^{Max}(\mathcal{M}_{NA})^3(V_1, t_1)(\mathcal{H}_{NA})(V_1, t_1)(V_1 - E_{Na}) \\ &+ g_L(V_1 - E_L) \\ I_E(t_3) &= g_K^{Max}(\mathcal{M}_K)^4(V_3, t_3)(V_3 - E_K) \\ &+ g_{Na}^{Max}(\mathcal{M}_{NA})^3(V_3, t_3)(\mathcal{H}_{NA})(V_3, t_3)(V_3 - E_{Na}) \\ &+ g_L(V_3 - E_L) \end{aligned}$$

From Figure 10 and Figure 11, we see that for typical action potential simulation responses, we have $m^3(V_1, t_1)h(V_1, t_1) \approx 0.35$ and $n^4(V_1, t_1) \approx 0.2$. Further, $m^3(V_3, t_3)h(V_3, t_3) \approx 0.01$ and $n^4(V_3, t_3) \approx 0.4$.

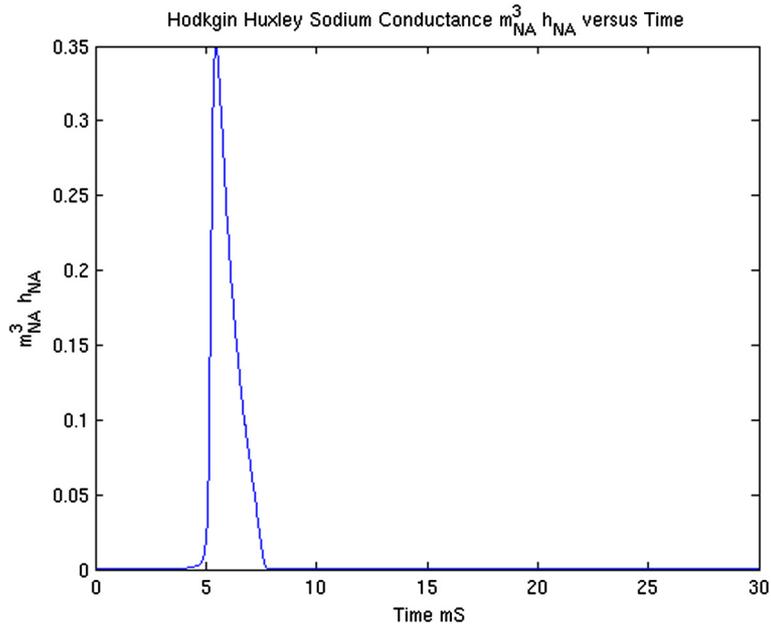


Figure 10 The product $\mathcal{M}_{NA}^3 \mathcal{H}_{NA}$: Sodium Conductances during a Pulse 1100 at 3.0.

Thus,

$$I_E(t_1) = 0.20g_K^{Max} (V_1 - E_K) + 0.35g_{Na}^{Max} (V_1 - E_{Na}) + g_L(V_1 - E_L)$$

$$I_E(t_3) = 0.40g_K^{Max} (V_3 - E_K) + 0.01g_{Na}^{Max} (V_3 - E_{Na}) + g_L(V_3 - E_L)$$

Reorganizing,

$$I_E(t_1) = (0.20g_K^{Max} + 0.35g_{Na}^{Max} + g_L) V_1 - (0.20g_K^{Max} E_K + 0.35g_{Na}^{Max} E_{Na} + g_L E_L)$$

$$I_E(t_3) = (0.40g_K^{Max} + 0.01g_{Na}^{Max} + g_L) V_3 - (0.40g_K^{Max} E_K + 0.01g_{Na}^{Max} E_{Na} + g_L E_L)$$

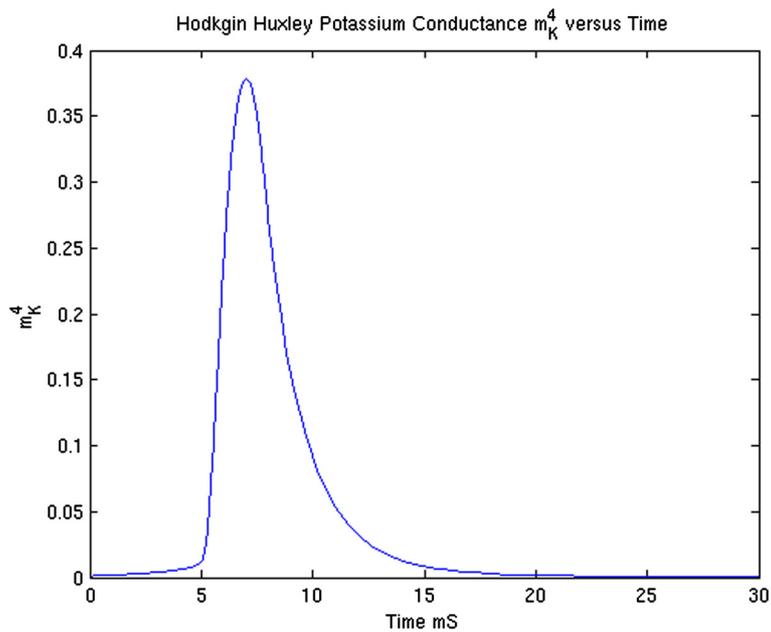


Figure 11 The product \mathcal{M}_K^4 : Potassium Conductances during a Pulse 1100 at 3.0.

Solving for the voltages, we find

$$V_1 = \frac{I_E(t_1) + 0.20g_K^{Max}E_K + 0.35g_{Na}^{Max}E_{Na} + g_L E_L}{0.20g_K^{Max} + 0.35g_{Na}^{Max} + g_L}$$

$$V_3 = \frac{I_E(t_3) + 0.40g_K^{Max}E_K + 0.01g_{Na}^{Max}E_{Na} + g_L E_L}{0.40g_K^{Max} + 0.01g_{Na}^{Max} + g_L}$$

Thus,

$$\frac{\partial V_1}{\partial g_K^{Max}} = 0.20E_K \frac{1}{0.20g_K^{Max} + 0.35g_{Na}^{Max} + g_L} + \frac{I_E(t_1) + 0.20g_K^{Max}E_K + 0.35g_{Na}^{Max}E_{Na} + g_L E_L}{0.20g_K^{Max} + 0.35g_{Na}^{Max} + g_L} \frac{-1.0}{0.20g_K^{Max} + 0.35g_{Na}^{Max} + g_L} .20$$

This simplifies to

$$\frac{\partial V_1}{\partial g_K^{Max}} = \frac{0.20}{0.20g_K^{Max} + 0.35g_{Na}^{Max} + g_L} (E_K - V_1) \quad (18)$$

Similarly, we find

$$\frac{\partial V_1}{\partial g_{Na}^{Max}} = \frac{0.35}{0.20g_K^{Max} + 0.35g_{Na}^{Max} + g_L} (E_{Na} - V_1) \quad (19)$$

$$\frac{\partial V_3}{\partial g_K^{Max}} = \frac{0.40}{0.40g_K^{Max} + 0.01g_{Na}^{Max} + g_L} (E_K - V_3) \quad (20)$$

$$\frac{\partial V_3}{\partial g_{Na}^{Max}} = \frac{0.40}{0.40g_K^{Max} + 0.01g_{Na}^{Max} + g_L} (E_{Na} - V_3) \quad (21)$$

We also know that as t goes to infinity, the action potential flattens and V'_m approaches 0. Also, the applied current, I_E is zero and so we must have

$$0 = -g_K^{Max}(\mathcal{M}_K)^4(V_\infty, \infty)(V_\infty - E_K) - g_{Na}^{Max}(\mathcal{M}_{NA})^3(V_\infty, \infty)(\mathcal{H}_{NA})(V_\infty, \infty)(V_\infty - E_{Na}) - g_L(V_\infty - E_L)$$

Our hyperpolarization model is

$$Y(t) = V_3 + (V_4 - V_3) \tanh(g(t - t_3))$$

We have V_∞ is V_4 . Thus,

$$0 = -g_K^{Max}(\mathcal{M}_K)^4(V_4, \infty)(V_4 - E_K) - g_{Na}^{Max}(\mathcal{M}_{NA})^3(V_4, \infty)(\mathcal{H}_{NA})(V_4, \infty)(V_4 - E_{Na}) - g_L(V_4 - E_L)$$

This gives, letting $(\mathcal{M}_{NA})^3(V_4, \infty)(\mathcal{H}_{NA})(V_4, \infty)$ and $(\mathcal{M}_K)^4(V_4, \infty)$ be denoted by $((\mathcal{M}_{NA})^3(\mathcal{H}_{NA}))^*$ and $((\mathcal{M}_K)^4)^*$ for simplicity of exposition,

$$\left(g_K^{Max} ((\mathcal{M}_K)^4)^* + g_{Na}^{Max} ((\mathcal{M}_{NA})^3(\mathcal{H}_{NA}))^* + g_L \right) V_4 = \left(g_K^{Max} ((\mathcal{M}_K)^4)^* E_K + g_{Na}^{Max} ((\mathcal{M}_{NA})^3(\mathcal{H}_{NA}))^* E_{Na} + g_L E_L \right)$$

Hence, letting $((\mathcal{M}_{NA})^3(\mathcal{H}_{NA}))^* \equiv (m^3h)^*$ and $((\mathcal{M}_K)^4)^* \equiv (n^4)^*$, we have

$$V_4 = \frac{g_K^{Max} n^4(V_4, \infty) E_K + g_{Na}^{Max} m^3(V_4, \infty) h(V_4, \infty) E_{Na} + g_L E_L}{g_K^{Max} n^4(V_4, \infty) + g_{Na}^{Max} m^3(V_4, \infty) h(V_4, \infty) + g_L}$$

We see

$$\frac{\partial V_4}{\partial g_K^{Max}} = \frac{n^4(V_4, \infty)}{g_K^{Max} n^4(V_4, \infty) + g_{Na}^{Max} m^3(V_4, \infty) h(V_4, \infty) + g_L} \times (E_K - V_4) \quad (22)$$

$$\frac{\partial V_4}{\partial g_{Na}^{Max}} = \frac{m^3(V_4, \infty) h(V_4, \infty)}{g_K^{Max} n^4(V_4, \infty) + g_{Na}^{Max} m^3(V_4, \infty) h(V_4, \infty) + g_L} \times (E_{Na} - V_4) \quad (23)$$

We can also assume that the area under the action potential curve from the point (t_0, V_0) to (t_1, V_1) is proportional to the incoming current applied. If V_{In} is the axon - hillock voltage, the impulse current applied to the axon - hillock is $g_{In} V_{In}$ where g_{In} is the ball stick model conductance for the soma. Thus, the approximate area under the action potential curve must match this applied current. We have $\frac{1}{2} (t_1 - t_0) (V_1 - V_0) \approx g_{In} V_{In}$ We conclude $(t_1 - t_0) = \frac{2g_{In} V_{In}}{V_1 - V_0}$. Thus

$$\frac{\partial(t_1 - t_0)}{\partial g_K^{Max}} = -\frac{t_1 - t_0}{V_1 - V_0} \frac{\partial V_1}{\partial g_K^{Max}} \quad \text{and} \quad (24)$$

$$\frac{\partial(t_1 - t_0)}{\partial g_{Na}^{Max}} = -\frac{t_1 - t_0}{V_1 - V_0} \frac{\partial V_1}{\partial g_{Na}^{Max}}$$

Also, we know that during the hyperpolarization phase, the sodium current is off and the potassium current is slowly bringing the membrane potential back to the reference voltage. Now, our BFV model does not assume that the membrane potential returns to the reference level. Instead, by using

$$Y(t) = V_3 + (V_4 - V_3) \tanh(g(t - t_3))$$

we assume the return is to voltage level V_4 . At the midpoint, $Y = \frac{1}{2}(V_3 + V_4)$, we find

$$\frac{1}{2}(V_4 - V_3) = (V_4 - V_3) \tanh(g(t - t_3))$$

Thus, letting $u = g(t - t_3)$, $\frac{1}{2} = \frac{e^{2u}-1}{e^{2u}+1}$ and we find $u = \frac{\ln(3)}{2}$. Solving for t , we then have $t^* = t_3 + \frac{\ln(3)}{2g}$. From t_3 on, the Hodgkin - Huxley dynamics are

$$C_m \frac{dV_M}{dt} = -g_K^{Max} (\mathcal{M}_K)^4 (V_m, t) (V_m - E_K) - g_L (V_m - E_L).$$

We want the values of the derivatives to match at t^* . This gives

$$g(V^* - V_3) \operatorname{sech}^2(g(t^* - t_3)) = -\frac{g_K^{Max}}{C_m} (\mathcal{M}_K)^4 (V^*, t^*) (V^* - E_K) - g_L (V^* - E_L)$$

where $V^* = \frac{1}{2}(V_3 + V_4)$. Now $g(t^* - t_3) = \frac{\ln(3)}{2}$ and thus we find

$$\begin{aligned} \frac{g}{2} (V_4 - V_3) \operatorname{sech}^2(g(t^* - t_3)) &= \\ -\frac{g_K^{Max}}{C_m} (\mathcal{M}_K)^4 (V^*, t) (V^* - E_K) - \frac{g_L}{C_m} (V^* - E_L) & \\ \frac{g}{2} (V_4 - V_3) \frac{9}{64} &= \\ -\frac{g_K^{Max}}{C_m} (\mathcal{M}_K)^4 (V^*, t^*) (V^* - E_K) - g_L (V^* - E_L) & \end{aligned}$$

Next, consider the magnitude of $(\mathcal{M}_K)^4 (V^*, t^*)$. We know at t^* , $(\mathcal{M}_K)^4$ is small from Figure 11. Thus, we will replace it by the value 0.01. This gives

$$\begin{aligned} \frac{g}{2} (V_4 - V_3) \frac{9}{64} &= -0.01 \frac{g_K^{Max}}{C_m} \left(\frac{1}{2} (V_4 + V_3) - E_K \right) \\ &\quad - \frac{g_L}{C_m} \left(\frac{1}{2} (V_4 + V_3) - E_L \right) \end{aligned}$$

Simplifying, we have

$$\begin{aligned} \frac{9g}{128} (V_4 - V_3) &= \left(0.01 \frac{g_K^{Max}}{C_m} E_K + \frac{g_L}{C_m} E_L \right) \\ &\quad - \frac{1}{2} \left(0.01 \frac{g_K^{Max}}{C_m} + \frac{g_L}{C_m} \right) (V_4 + V_3) \\ \frac{9g}{64} &= \left(0.01 \frac{g_K^{Max}}{C_m} E_K + \frac{g_L}{C_m} E_L \right) \frac{1}{V_4 - V_3} \\ &\quad - \left(0.01 \frac{g_K^{Max}}{C_m} + \frac{g_L}{C_m} \right) \frac{V_4 + V_3}{V_4 - V_3} \end{aligned}$$

We can see clearly from the above equation, that the dependence of g on g_K^{Max} and g_{Na}^{Max} is quite complicated. However, we can estimate this dependence as follows. We know that $V_3 + V_4$ is about the reference

voltage, -65.9mV . If we approximate V_3 by the potassium battery voltage, $E_K = -72.7\text{mV}$ and V_4 by the reference voltage, we find $\frac{V_3+V_4}{V_4-V_3} \approx \frac{-138.6}{6.8} = -20.38$ and $\frac{1}{V_4-V_3} \approx \frac{1}{6.8} = 0.147$. Hence,

$$\begin{aligned} \frac{9C_m g}{64} &= 0.147 (0.01 g_K^{Max} E_K + g_L E_L) \\ &\quad + 20.38 (0.01 g_K^{Max} + g_L) \\ &= (0.0147 E_K + 2.038 E_L) g_K^{Max} \\ &\quad + g_L (0.0147 E_L + 20.38) \end{aligned}$$

Thus, we find

$$\frac{\partial g}{\partial g_K^{Max}} = \frac{64}{9C_m} (0.0147 E_K + 2.038 E_L) \quad (25)$$

This gives $\frac{\partial g}{\partial g_K^{Max}} \approx -710.1$. Equation 25 shows what our intuition tells us: if g_K^{Max} increases, the potassium current is stronger and the hyperpolarization phase is shortened. On the other hand, if g_K^{Max} decreases, the potassium current is weaker and the hyperpolarization phase is lengthened.

Multiple inputs

Consider a typical input $V(t)$ which is determined by a BFV vector. Without loss of generality, we will focus on excitatory inputs in our discussions. The input consists of a three distinct portions. First, a parabolic cap above the equilibrium potential determined by the values (t_0, V_0) , (t_1, V_1) , (t_2, V_2) . Next, the input contains half of another parabolic cap dropping below the equilibrium potential determined by the values (t_2, V_2) and (t_3, V_3) . Finally, there is the hyperpolarization phase having functional form $H(t) = V_3 + (V_4 - V_3) \tanh(g(t - t_3))$. Now assume two inputs arrive at the same electronic distance L . We label this inputs as A and B as is shown in Figure 12.

For convenience of exposition, we also assume $t_3^A < t_3^B$, as otherwise, we just reverse the roles of the variables in our arguments. In this figure, we note only the minimum points on the A and B curves. We merge these inputs into a new input V^N prior to the hyperpolarization phase as follows:

$$\begin{aligned} t_0^N &= \frac{t_0^A + t_0^B}{2}, \quad V_0^N = \frac{V_0^A + V_0^B}{2}, \quad \text{and } t_1^N = \frac{t_1^A + t_1^B}{2} \\ V_1^N &= \frac{V_1^A + V_1^B}{2}, \quad t_2^N = \frac{t_2^A + t_2^B}{2}, \quad \text{and } V_2^N = \frac{V_2^A + V_2^B}{2} \end{aligned}$$

This constructs the two parabolic caps of the new resultant input by averaging the caps of V^A and V^B . The construction of the new hyperpolarization phase is more complicated. The shape of this portion of an action potential has a profound effect on neural modulation, so it is

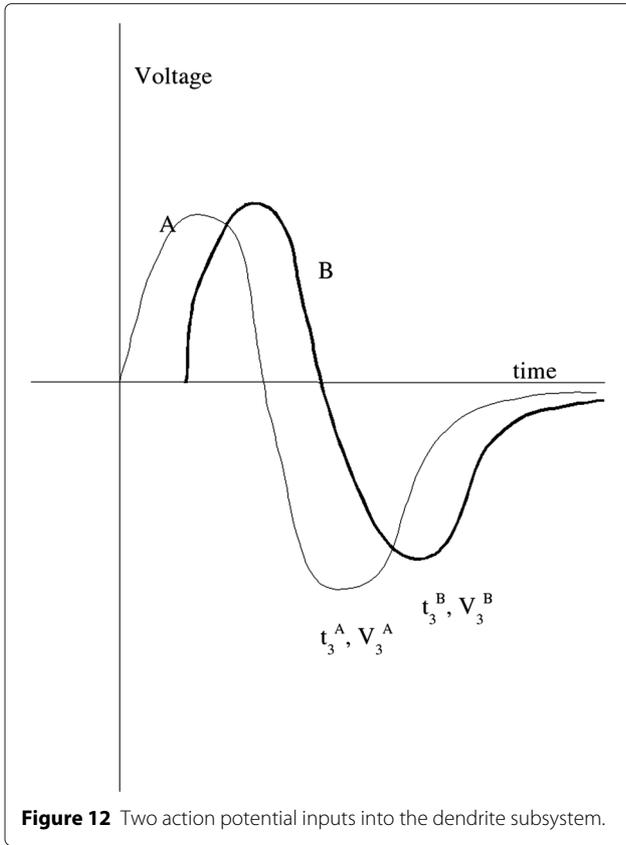


Figure 12 Two action potential inputs into the dendrite subsystem.

very important to merge the two inputs in a reasonable way. The hyperpolarization phases of V^A and V^B are given by

$$H^A(t) = V_3^A + (V_4^A - V_3^A) \tanh(g^A(t - t_3^A))$$

$$H^B(t) = V_3^B + (V_4^B - V_3^B) \tanh(g^B(t - t_3^B))$$

We will choose the 4 parameters V_3, V_4, g, t_3 so as to minimize

$$E = \int_{t_3^A}^{\infty} (H(t) - H^A(t))^2 + (H(t) - H^B(t))^2 dt$$

For optimality, we find the parameters where $\frac{\partial E}{\partial V_3}, \frac{\partial E}{\partial V_4}, \frac{\partial E}{\partial g}$ and $\frac{\partial E}{\partial t_3}$ are 0. Now,

$$\frac{\partial E}{\partial V_3} = \int_{t_3^A}^{\infty} 2 \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \frac{\partial H}{\partial V_3} dt$$

Further,

$$\frac{\partial H}{\partial V_3} = 1 - \tanh(g(t - t_3))$$

so we obtain

$$0 = \int_{t_3^A}^{\infty} 2 \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \times (1 - \tanh(g(t - t_3))) dt \tag{26}$$

We also find

$$\frac{\partial E}{\partial V_4} = \int_{t_3^A}^{\infty} 2 \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \times (\tanh(g(t - t_3))) dt$$

as

$$\frac{\partial H}{\partial V_4} = \tanh(g(t - t_3))$$

The optimality condition then gives

$$0 = \int_{t_3^A}^{\infty} 2 \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \tanh(g(t - t_3)) dt \tag{27}$$

Combining equation 26 and equation 27, we find

$$0 = \int_{t_3^A}^{\infty} \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \tanh(g(t - t_3)) dt.$$

It follows after simplification, that

$$0 = \int_{t_3^A}^{\infty} \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} dt \tag{28}$$

The remaining optimality conditions give

$$\frac{\partial E}{\partial g} = \int_{t_3^A}^{\infty} 2 \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \frac{\partial H}{\partial g} dt = 0$$

$$\frac{\partial E}{\partial t_3} = \int_{t_3^A}^{\infty} 2 \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \frac{\partial H}{\partial t_3} dt = 0$$

We calculate

$$\frac{\partial H}{\partial g} = (V_4 - V_3)(t - t_3) \operatorname{sech}^2(g(t - t_3))$$

$$\frac{\partial H}{\partial t_3} = -(V_4 - V_3)g \operatorname{sech}^2(g(t - t_3))$$

Thus, we find

$$0 = \int_{t_3^A}^{\infty} \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \times (V_4 - V_3)(t - t_3) \operatorname{sech}^2(g(t - t_3)) dt$$

$$0 = \int_{t_3^A}^{\infty} \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \times (V_4 - V_3) g \operatorname{sech}^2(g(t - t_3)) dt.$$

This implies

$$0 = \int_{t_3^A}^{\infty} \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} t \operatorname{sech}^2(g(t - t_3)) dt$$

$$- t_3 \int_{t_3^A}^{\infty} \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \operatorname{sech}^2(g(t - t_3)) dt$$

$$0 = \int_{t_3^A}^{\infty} \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} \operatorname{sech}^2(g(t - t_3)) dt.$$

This clearly can be simplified to

$$0 = \int_{t_3^A}^{\infty} \left\{ (H(t) - H^A(t)) + (H(t) - H^B(t)) \right\} t \operatorname{sech}^2(g(t - t_3)) dt \tag{29}$$

We can then satisfy equation 28 and equation 29 by making

$$(H(t) - H^A(t)) + (H(t) - H^B(t)) = 0. \tag{30}$$

Equation 30 can be rewritten as

$$0 = \left(V_3 - \frac{V_3^A + V_3^B}{2} \right) + (V_4 - V_3) \tanh(g(t - t_3))$$

$$- \frac{V_4^B - V_3^B}{2} \tanh(g^B(t - t_3^B)) - \frac{V_4^A - V_3^A}{2} \tanh(g^A(t - t_3^A)) \tag{31}$$

This equation is true as $t \rightarrow \infty$. Thus, we obtain the identity

$$0 = \left(V_3 - \frac{V_3^A + V_3^B}{2} \right) + (V_4 - V_3) - \frac{V_4^B - V_3^B}{2}$$

$$- \frac{V_4^A - V_3^A}{2}$$

Upon simplification, we find $0 = V_3 - \frac{V_3^A + V_3^B}{2}$ and $0 = V_4 - \frac{V_4^A + V_4^B}{2}$. This leads to our choices for V_3 and V_4 . $V_3 =$

$\frac{V_3^A + V_3^B}{2}$ and $V_4 = \frac{V_4^A + V_4^B}{2}$ Equation 31 is also true at $t = t_3^A$ and $t = t_3^B$. This gives

$$0 = \left(V_3 - \frac{V_3^A + V_3^B}{2} \right) + (V_4 - V_3) \tanh(g(t_3^A - t_3))$$

$$- \frac{V_4^B - V_3^B}{2} \tanh(g^B(t_3^A - t_3^B)) \tag{32}$$

$$0 = \left(V_3 - \frac{V_3^A + V_3^B}{2} \right) + (V_4 - V_3) \tanh(g(t_3^B - t_3))$$

$$- \frac{V_4^A - V_3^A}{2} \tanh(g^A(t_3^B - t_3^A)) \tag{33}$$

For convenience, define $w_{34}^A = \frac{V_4^A - V_3^A}{2}$ and $w_{34}^B = \frac{V_4^B - V_3^B}{2}$. Then, using equation 32 and equation 33, we find

$$0 = (V_4 - V_3) \tanh(g(t_3^A - t_3))$$

$$- w_{34}^B \tanh(g^B(t_3^A - t_3^B))$$

$$0 = (V_4 - V_3) \tanh(g(t_3^B - t_3))$$

$$- w_{34}^A \tanh(g^A(t_3^B - t_3^A))$$

This is then rewritten as

$$\tanh(g(t_3^A - t_3)) = \frac{w_{34}^B \tanh(g^B(t_3^A - t_3^B))}{(V_4 - V_3)}$$

$$\tanh(g(t_3^B - t_3)) = \frac{w_{34}^A \tanh(g^A(t_3^B - t_3^A))}{(V_4 - V_3)}$$

Defining

$$z_A = \frac{w_{34}^B \tanh(g^B(t_3^A - t_3^B))}{V_4 - V_3} \text{ and}$$

$$z_B = \frac{w_{34}^A \tanh(g^A(t_3^B - t_3^A))}{V_4 - V_3}$$

we find that the optimality conditions have led to the two nonlinear equations for g and t_3 given by

$$\tanh(g(t_3^A - t_3)) = z_A \text{ and} \tag{34}$$

$$\tanh(g(t_3^B - t_3)) = z_B$$

Note we thus have

$$z_A = \frac{w_{34}^B \tanh(g^B(t_3^A - t_3^B))}{V_4 - V_3} =$$

$$- \frac{w_{34}^B \tanh(g^B(t_3^B - t_3^A))}{w_{34}^A + w_{34}^B}$$

$$z_B = \frac{w_{34}^A \tanh(g^A(t_3^B - t_3^A))}{V_4 - V_3}$$

$$= \frac{w_{34}^A \tanh(g^A(t_3^A - t_3^B))}{w_{34}^A + w_{34}^B}$$

Hence,

$$z_A > -\frac{w_{34}^B}{w_{34}^A + w_{34}^B} > -1 \text{ and } z_B < \frac{w_{34}^A}{w_{34}^A + w_{34}^B} < 1$$

so that $z_A < 0 < z_B$. It seems reasonable that the optimal value of t_3 should lie between t_3^A and t_3^B . Note equations 34 precludes the solutions $t_3 = t_3^A$ or $t_3 = t_3^B$. To solve the nonlinear system for g and t_3 , we will approximate \tanh by its first order Taylor Series expansion. This seems reasonable as we don't expect $g(t_3^A - t_3)$ and $g(t_3^B - t_3)$ to be far from 0. This gives the approximate system

$$g(t_3^A - t_3) \approx z_A \text{ and } g(t_3^B - t_3) \approx z_B \quad (35)$$

Using these, we find $g = \frac{z_B}{t_3^B - t_3}$ and we obtain $\frac{z_B}{t_3^B - t_3} (t_3^A - t_3) = z_A$. This can be simplified as follows:

$$\frac{t_3^A - t_3}{t_3^B - t_3} = \frac{z_A}{z_B}, \quad (t_3^A - t_3) z_B = (t_3^B - t_3) z_A \text{ and}$$

$$t_3^A z_B - t_3^B z_A = t_3 (z_B - z_A)$$

Thus, we find the optimal value of t_3 is approximately

$$t_3 = \frac{t_3^A z_B - t_3^B z_A}{z_B - z_A} \quad (36)$$

Using the approximate value of t_3 , we find the optimal value of g can be approximated as follows:

$$g = \frac{z_B}{t_3^B - \frac{t_3^A z_B - t_3^B z_A}{z_B - z_A}} = \frac{z_B(z_B - z_A)}{t_3^B(z_B - z_A) - (t_3^A z_B - t_3^B z_A)}$$

$$= \frac{z_B(z_B - z_A)}{t_3^B z_B - t_3^A z_B} = \frac{z_B - z_A}{t_3^B - t_3^A}$$

Hence, we find the approximate optimal value of g is

$$g = \frac{z_B - z_A}{t_3^B - t_3^A} \quad (37)$$

It is easy to check that this value of t_3 lies in (t_3^A, t_3^B) as we suspected it should and that g is positive. We summarize our results. Given two input BFVs, the sigmoid portions of the incoming BFVs combine into the new sigmoid given by

$$H(t) = V_3 + (V_4 - V_3) \tanh(g(t - t_3))$$

$$H(t) = \frac{V_3^A + V_3^B}{2} + \left(\frac{V_4^A - V_3^A}{2} + \frac{V_4^B - V_3^B}{2} \right) \tanh \left(\frac{z_B - z_A}{t_3^B - t_3^A} \left(t - \frac{t_3^A z_B - t_3^B z_A}{z_B - z_A} \right) \right)$$

Given an input sequence of BFV's into a port on the dendrite of an accepting neuron $\{V_n, V_{n-1}, \dots, V_1\}$ the procedure discussed above computes the combined response

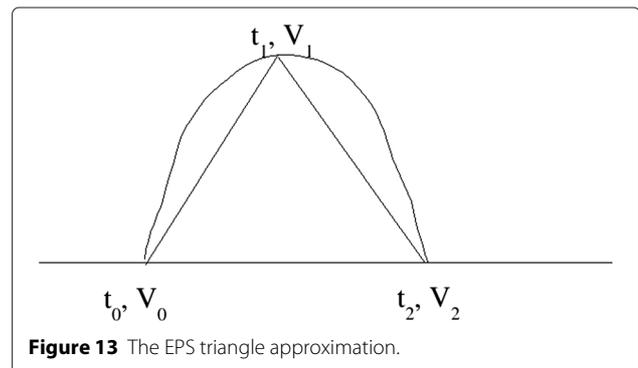
that enters that port at a particular time. The inputs into the dendritic system are combined pairwise; V_2 and V_1 combine into a V_{new} which then combines with V_3 and so on. We can do this at each electrotonic location.

Pre-neurons can supply input to the dendrite cable at electronic positions $w = 0$ to $w = 4$. These inputs generate an ESP or ISP via many possible mechanisms or they alter the structure of the dendrite cable itself by the transcription of proteins. The output of a pre-neuron is a BFV which must then be associated with an abstract trigger as we have discussed in earlier chapters. The strength of a BFV output will be estimated as follows: The area under the first parabolic cap of the BFV can be approximated by the area of the triangle, A , formed by the vertices $(t_0, V_0), (t_1, V_1), (t_2, V_2)$. This area is shown in Figure 13. The area is given by $A = \frac{1}{2}(V_2 - V_0)(t_2 - t_0)$.

The size of this area then allows us to determine the first and second messenger contributions this input makes to the post-neuron.

Results and discussion

In Erlang, (Armstrong 2013), the fundamental unit of calculation in the **process**. Each process does not share information with any other process and hence, the memory model followed is based on message passing. There are other approaches such as shared memory with locks, software transactional memory (used in Clojure) and futures and promises. We focus on Erlang's model as we feel a neuron in a brain model is a computational unit which performs a calculation on the basis of current inputs (i.e. messages) and sends its output to other nodes based on its forward link set. In Erlang, the message passing is asynchronous (as is the case in the BFV outputs a brain model's node sends out to its recipients) and hence data must be copied for the message. Hence, since we want to have many node and edge functions in the brain model, $\mathcal{G}(N, E)$, it is important to keep the amount of calculation in each node and edge low. To achieve that, we need to *approximate* the neuronal calculations as we have been discussing here. Instead of solving systems of ordinary



or partial differential equations are each node, we have found approximations to the solutions which capture a reasonable amount of the information processing complexity that is there. Also, the approach we use allows us to pay close attention to the architectural skeleton given by the graph $\mathcal{G}(\mathcal{N}, \mathcal{E})$ and separate the processing function from it. Another approach is to use a middle ware tool such as **CORBA** implemented in C++ using **omniORB-4.1.4** to handle multiple core/ multiple CPU models, but the functional programming approach using **Erlang** with no shared memory is better suited to this task. In addition, we want the model to be able to use both first and second messenger triggers even though the second messenger inputs alter the computational abilities of the node they effect. We handle this by allowing the second messenger triggers to alter the low dimensional BFV. To indicate how this would be done, let's consider a model with neurons that are first messenger types (i.e. alter the sodium and potassium gate parameters in our simple standard Hodgkin-Huxley model) and two additional neuron types that generate second messenger triggers such as dopamine and serotonin. Let's call these classes \mathcal{N}_F , \mathcal{N}_D and \mathcal{N}_S . We are aware the situation is much more complicated but this simple thought experiment will show how our approximations are used. Consider a node $\mathcal{N} = \mathcal{y}_i$ in the graph and recall we are using the general update strategy given by

$$\mathcal{y}_i(t + 1) = \mathcal{I}_i + \sum_{j \in \mathcal{B}(i)} \mathcal{E}_{j \rightarrow i}(t) \mathcal{Y}_j(t) \text{ where}$$

$$\mathcal{Y}_i(t + 1) = \sigma_i(t)(\mathcal{y}_i(t)).$$

where each σ represents a nodal process and each $\mathcal{E}_{j \rightarrow i}$ an edge process as modeled in Erlang with \mathcal{y} denoting the input to a node and \mathcal{Y} , the output from the node.

- When the process that handles this node looks at its input queue, it sees messages \mathcal{M} organized into the families \mathcal{M}_F , \mathcal{M}_D and \mathcal{M}_S . We use the combining inputs algorithm to merge multiple inputs in each message class into one input which we will denote by \mathcal{I}_F , \mathcal{I}_D and \mathcal{S} .
- The inputs \mathcal{I}_F , \mathcal{I}_D and \mathcal{S} then generate an trigger update using the EPS/ IPS triangle approximation. given by \mathbf{a}_F , \mathbf{a}_D , and \mathbf{a}_S . The latter two are second messengers. Recall, a second messenger trigger T creates activated kinase PK/\mathcal{U} and each unit of T creates λ units of PK/\mathcal{U} where λ is quite large – perhaps 10,000 or more times the base level of $[PK/\mathcal{U}]_e$. Thus, letting $r_{PK/\mathcal{U}} = \beta$ and $\mathcal{H} = \frac{k_1 k_2 k_3}{k_{-1} k_{-2} k_{-3}}$, we know $\delta_T = (2\beta + \beta^2) \mathcal{H} [PK/\mathcal{U}]_e^2$. Denote the actual trigger updates by δ_F , δ_D , and δ_S . Hence, we can

model each of the two second messenger trigger changes as

$$\delta_D \propto (2\beta_D + \beta_D^2) \mathbf{a}_D \text{ and } \delta_S \propto (2\beta_S + \beta_S^2) \mathbf{a}_S.$$

From our discussion of how the BFV is altered by triggers given in Section “Modulation of the BFV parameters”, it is clear a second messenger trigger update initiates changes in the BFV. The literature on the effects a neurotransmitter has on the action potential of an excitable neuron gives us specific information about what parts of the action potential are changed. We do not discuss these details here for brevity. Suffice it to say, we can assign each neurotransmitter to a BVF alteration. Thus, letting the proportionality constants above be \mathcal{K}_D and \mathcal{K}_S

$$\delta_D = (2\beta_D + \beta_D^2) \mathbf{a}_D \mathcal{K}_D \implies \nabla_D(BFV)$$

$$\delta_S = (2\beta_S + \beta_S^2) \mathbf{a}_S \mathcal{K}_S \implies \nabla_S(BFV)$$

where the gradients here are the specific changes in the 11 parameters of the BFV that each neurotransmitter causes. Also, recall the efficacy of the second messenger release depends r_u , the rate of re-uptake in the connection between two nodes, the second messenger destruction rate r_d , the rate of second messenger release, r_r and the density of the second messenger receptor, n_d . The triple (r_u, r_d, r_r) thus determines a net increase or decrease of second messenger concentration between two nodes: $r_r - r_u - r_d \equiv r_{net}$. The efficacy of a connection between nodes is then proportional to the product $r_{net} \times n_d$. The density of the second messenger receptor is amenable to second messenger alterations via triggers as well. If we change our update equations to

$$\delta_D = (r_r - r_u - r_d) (2\beta_D + \beta_D^2) \mathbf{a}_D \mathcal{K}_D$$

$$\delta_S = (2\beta_S + \beta_S^2) \mathbf{a}_S \mathcal{K}_D.$$

by absorbing the n_d term into the proportionality constants, we have a mechanism that allows us to model neurotransmitter interaction in the synaptic cleft.

- For the input \mathcal{I}_F which is a first messenger type, we know this will cause a change in maximum sodium or potassium conductance and perhaps more complicated combinations. These were explored in Section “Modulation of the BFV parameters”. For example, we can estimate how the parameter V_1 of the BFV changes via

$$\frac{\partial V_1}{\partial g_{Na}^{Max}} = \frac{0.35}{0.20g_K^{Max} + 0.35g_{Na}^{Max} + g_L} (E_{Na} - V_1)$$

This tells us

$$\delta V_1 = \left(\frac{0.35}{0.20g_K^{Max} + 0.35g_{Na}^{Max} + g_L} (E_{Na} - V_1) \right) \delta g_{Na}^{Max}.$$

Then, from Section “Second messengers”, we know

$$\delta g_{Na}(t, V) = g_{Na}^{max}(e\delta_{Na} \sigma([P(T_1)], 0, g_{Na})) \mathcal{M}_{Na}^p(t, V) \mathcal{H}_{Na}^q(t, V)$$

which we can approximate using the input \mathbf{a}_F . Thus, $\delta g_{Na}(t, V) \propto \mathbf{a}_F$ with proportionality constant \mathbf{K}_F . Thus,

$$\delta V_1 = \left(\frac{0.35}{\mathbf{K}_F \mathbf{a}_F} (E_{Na} - V_1) \right)$$

is a reasonable approximation to this alteration of the BFV due to this first messenger input.

Hence, our processing node handles the messages in its input queue using simple arithmetic and a few stored parameters. Each edge process computes the r_{net} term required but it is still simpler as it only has to connect the BFV from one neuron to another.

Conclusions

We have shown how to approximate neuronal computation for both first and second messenger systems so that a graph model $\mathcal{G}(N, E)$ can be implemented efficiently in a modern functional programming language such as Erlang. Other languages are possible but our focus was on Erlang alone here. The simulation of a brain model then can take advantage of as many cores as are available on our hardware. **Erlang** is not designed for heavy computation, so this is why we have spent so much time discussing ways to approximate neural computations at each node and the synaptic processing. It is important to note that new and interesting simulations require us to pay much closer attention to the actual hardware we will be using. Hence, while the details of the multicore use might change with new hardware, the basic principles of how we combine computational algorithms to hardware via software will be retained.

Competing interests

The author declares that he has no competing interests.

Authors' contribution

JP performed all the research for this work. The author has read and approved the final manuscript.

Received: 4 September 2014 Accepted: 6 April 2015

Published online: 07 July 2015

References

- Armstrong, J (2013). *Programming Erlang Second Edition: Software for a Concurrent World*. Dallas, TX: The Pragmatic Bookshelf.
- Bray, D (1998). Signalling complexes: Biophysical constraints on intracellular communication. *Annual Review of Biophysics and Biomolecular Structure*, 27, 59–75.
- Friston, K (2005). A theory of cortical responses. *Philosophical Transactions of the Royal Society B*, 360, 815–836.
- Friston, K (2010). The free-energy principle: a unified brain theory? *Nature Reviews: Neuroscience*, 11, 127–138.
- Gerhart, J, & Kirschner, M (1997). *Cells, Embryos and Evolution: Towards a Cellular and Developmental Understanding of Phenotypic Variation and Evolutionary Adaptability*. USA: Blackwell Science.
- Hille, B (1992). *Ionic Channels of Excitable Membranes*. New York: Sinauer Associates Inc.
- Li, X, Xia, S, Bertish, H, Branch, C, DeLisi, L (2012). Unique topology of language processing brain network: A systems-level biomarker of schizophrenia. *Schizophrenia Research*, 141, 128–136.
- Maia, T, & Frank, M (2011). From reinforcement learning models to psychiatric and neurological disorders. *Nature Neuroscience*, 14(2), 154–162.
- Peterson, J (2014a). Computation in networks. *Computational Cognitive Science*. This issue.
- Peterson, J (2014b). *Bioinformatics Processing, A Primer on Computational Cognitive Science*, In *Preparation, Cognitive Science and Technology Series*. Singapore: Springer.
- Peterson, J (2014c). *Calculus for Cognitive Scientists: Partial Differential Equation Models*, In *Preparation, Cognitive Science and Technology Series*. Singapore: Springer.
- Peterson, J, & Khan, T (2006). Abstract action potential models for toxin recognition. *Journal of Theoretical Medicine*, 6(4), 199–234.
- Russo, S, & Nestler, E (2013). The brain reward circuitry in mood disorders. *Nature Reviews: Neuroscience*, 14, 609–625.
- Sherman, S (2004). Interneurons and triadic circuitry of the thalamus. *Trends in Neuroscience*, 27(11), 670–675.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com