Analyzing a Discrete Model of *Aplysia* Central Pattern Generator*

Ashish Tiwari and Carolyn Talcott

SRI International, Menlo Park, CA 94025, {tiwari,clt}@csl.sri.com

Abstract. We present a discrete formal model of the central pattern generator (CPG) located in the buccal ganglia of *Aplysia* that is responsible for mediating the rhythmic movements of its foregut during feeding. Our starting point is the continuous dynamical model for pattern generation underlying fictive feeding in *Aplysia* proposed by Baxter et. al. [4]. The discrete model is obtained as a composition of discrete models of ten individual neurons in the CPG. The individual neurons are interconnected through excitatory and inhibitory synaptic connections and electric connections. We used Symbolic Analysis Laboratory (SAL) to formally build the model and analyzed it using the SAL model checkers. Using abstract discrete models of the individual neurons helps in understanding the buccal motor programs generated by the network in terms of the network connection topology. It also eliminates the need for detailed knowledge of the unknown parameters in the continuous model of Baxter et. al. [4].

1 Introduction

Background The last several years have witnessed rapid growth in the amount of detailed high quality experimental data on neural processes underlying behavior. Concurrently, computational neuroscience has also experienced a surge of activity in the formulation of models of increasing complexity. These twin developments present opportunities as well as challenges to neuroinformatics. There are exciting opportunities to describe and simulate neural processes in hitherto unprecedented detail. The challenges are to manage vast amounts of intricate data consisting of experimental and model-derived results, and also to construct tractable models at levels of abstraction that provide useful insights for guiding research. Many current models of neural processes utilize a framework of differential equations such as the Hodgkin-Huxley (H-H) equations [2]. Other modeling techniques include integrate-and-fire methods and artificial neural networks [1, 12]. Detailed models, such as the H-H models, tend to exhibit high sensitivity to system parameters and consequently require very accurate measurements of these parameters. However, such data are frequently unavailable. Furthermore, these models do not scale easily because they rapidly become

* Research supported in part by the National Science Foundation under grants IIS-0513857 and CNS-0720721.

intractable as the number of cells incorporated increases. The situation is analogous to that in Systems Biology. A complementary approach based on using logical computing formalisms and symbolic techniques [5, 14, 19], called Symbolic Systems Biology, is being successfully pursued. By representing knowledge at an abstract, symbolic level this approach has enabled development of models capable of performing sophisticated queries about large models of biological processes, while not being overly constrained by lack of low-level details.

Neurons and Neural Circuits Neurons are highly specialized eukaryotic cells capable of communicating with each other by means of electrical and chemical signaling (see, for example, [3]). While there are several kinds of neurons, all possess a cell body, called the soma, from which emerge several tree like structures called dendrites as well as a single long tube called the axon, which ends in several branches, the synaptic terminals. The synaptic terminals of the transmitting (presynaptic) neuron communicate to the dendrites of nearby receiving (postsynaptic) neurons by releasing specialized molecules, the neurotransmitters, such as glutamate, serotonin, and dopamine. A neurotransmitter is released by the presynaptic neuron once a sufficient transmembrane voltage depolarization has reached the synaptic terminal. The buildup of membrane depolarization at the synapse occurs after a nerve impulse, or action potential, travels from the cell body along the axon in a series of depolarizations and repolarizations caused by transfer of sodium and potassium ions between the cytoplasm of the neuron and the extracellular space. The ions are transferred by electrochemical gradients and they cross the membrane through voltage-dependent channels, which are membrane pores that become permeable to ions due to conformational changes induced by transmembrane voltage depolarizations. Chemical synaptic connections between neurons can be of two types, inhibitory, and excitatory. Neurons can also signal each other via electrical synapses (also called gap junctions) in which the membranes of the two neurons contact each other via transmembrane pores through which electrical signals can spread bidirectionally. All of these, and other factors, endow neural signaling with a rich collection of signaling modalities and enormous complexity in communications and signaling behavior.

Feeding Behavior and its Neural Circuit A major goal of neuroscience is to understand neural processes that underlie the generation of behavior and the modification of behavior induced by learning. A basic tenet of neuroscience is that the ability of the nervous system to generate behaviors arises from the organization of neurons into networks or circuits and that the functional capabilities of these circuits emerge from interactions among: i) the intrinsic biophysical properties of the individual neurons, ii) the pattern of the synaptic connections among these neurons, and iii) the physiological properties of these synaptic connections. To investigate how neural circuits function, a diverse collection of animal model systems has been developed. By virtue of their relatively simple nervous systems, often with large identifiable neurons that are amenable to detailed study, invertebrates are frequent candidates for cellular analyses of neural circuits and their relationship to behavior. A key advantage of invertebrate model systems

is that the neural circuits controlling specific behaviors often contain relatively few neurons, which allows the circuit to be described in detail on a cell-by-cell and synapse-by-synapse basis. Another advantage is that many of these neural circuits produce fictive motor patterns when isolated in vitro, which facilitates investigating how behaviorally relevant neural activity is generated and regulated.

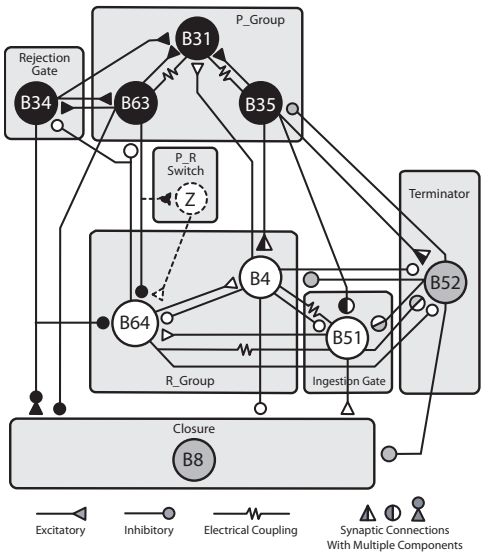
One useful animal model system is the marine mollusk Aplysia. Feeding behavior in Aplysia has been an important focus of study and research. The neural circuitry that mediates feeding behavior is located primarily in the cerebral and buccal ganglia. The structure of this circuit is relatively well understood [6]. A major component is the central pattern generator (CPG) in the buccal ganglia (Figure 1) that generates the motor activity for controlling the rhythmic movements of the odontophore and radula. Although a great deal is known about the individual components of the feeding circuitry, how this collection of cells and synapses functions as a circuit is not well understood. The nonlinear, diverse and dynamic nature of neural circuits provide formidable challenges to systematically analyzing and understanding how circuits operate and adapt. Symbolic system models can help in this endeavor. We believe that properties of a neuron network can be understood as properties of the *connections* between the neurons in the network. Such abstract models can also be used to test the plausibility of hypotheses and to manipulate components of the system in a manner that may not be feasible in the real nervous system.

2 Biology

We describe the biology of feeding in *Aplysia* and its neural control. The material here is taken mainly from Baxter et. al. [17, 4].

The feeding cycle in *Aplysia* consists of ingestion, which brings food into the buccal cavity, and egestion, which ejects unwanted material from the buccal cavity. These two stages of feeding involve rhythmic movements of structures in the foregut that can be classified into two phases: a protraction phase, where the jaws open and the odontophore rotate forward and a retraction phase, the odontophore retracts and the jaws close. During ingestion, the two halves of the radula (grasping surface of the odontophore) are separated and open during protraction, and closed during retraction. This causes the food to enter the buccal cavity. On the other hand, during egestion, the radula is closed during protraction and open during retraction. This causes ejection of unwanted material.

The buccal ganglia of *Aplysia* contain a central pattern generator (CPG) that controls the rhythmic movements of the foregut during feeding. The CPG generates two corresponding buccal motor programs (BMP), one for ingestion and another for egestion. Based on extensive intracellular recordings of action potentials, Baxter et.al. [17, 4] have proposed a speculative type of model of the CPG. This model contains continuous dynamical models of ten neuron cells: B4, B8, B31, B34, B35, B51, B52, B63, B64, and a hypothetical neuron Z, and their connections.



This diagram shows the model of the central pattern generator. The circles represent neurons and the edges represent the interconnections. Subsets of neurons are collected into groups to indicate their function/role. The P_Group neurons are active during protraction and the R_Group neurons during retraction. The rejection gate neuron participates in egestion, whereas the ingestion gate neuron plays a role in the ingestion BMP. The circle-terminated edge represents inhibitory synaptic connection, the triangle-terminated edge denotes an excitatory synaptic connection, and the wiggly edge denotes an electrical coupling. An edge terminating on a group indicates that it effects all members of the group.

Fig. 1. Central Pattern Generator: Neurons and their interconnections that are responsible for generating the BMPs associated with ingestion and egestion [4].

We give a brief description of the role of these neurons. B8 is a radula-closer motor neuron and the timing of its activity can distinguish ingestion from egestion. Neurons B31, B35, and B63 are active during the radula protraction phase whereas B4 and B64 are active during the radula retraction phase. The neuron B52 terminates the retraction phase. The hypothetical neuron Z mediates transition from the protraction to the retraction phase. The neurons B34 and B51 control B8 and are part of the system that switches mode from egestion to ingestion and vice-versa.

The ten identified neurons are connected to each other forming a complex network. Each interconnection is either an excitatory synaptic connection, an inhibitory synaptic connection, or an electric coupling. The connections between the ten neurons used in the model of Baxter et.al. [17, 4] are reproduced in Figure 1.

3 Related Work

Baxter et.al. [4] presented a continuous dynamical system model of the ten neuron interconnected network. They used continuous differential equations, in the form of Hodgkin-Huxley-type models, for each of the ten cells and their electrical and synaptic connections. Unfortunately, such a model requires inferring, either experimentally or computationally, several parameters that describe the details

of membrane currents. Specifically, there are about 18 parameters for each ion channel (Na, K) of each neuron. In addition, there are three parameters for each synaptic connection, two for synaptic plasticity, and two for every electric coupling. Estimating these parameters is a challenge and after the parameters have been discovered, the result is an immensely complicated model that can only be analyzed by simulation.

In [11] a simple rewriting logic model of a two neuron subsystem (B31,B63) of the Aplysia bucchal ganglia was presented. This work demonstrated that essential features of the two neuron system could be modeled by appropriate choice of a small number of parameters. This preliminary success lead us to look for a more principled way of determining the parameters that control a neuron's behavior. After studying the various proposed neuron models [12, 9], including the "simple model" of [12], we built a highly abstract qualitative model of a single neuron and used it as a starting point.

Hybrid systems have been used as a modeling language for System Biology [7] in general and for modeling single neurons [8] in particular. Due to their high expressiveness, hybrid models can more closely approximate HH models, but hybrid analysis does not scale to studying a large collection of cells and is restricted to studying a single cell [8]. Our interest was in analyzing a large neuron network and this motivated looking at discrete models.

4 Discrete Formal Model

In this section we present a simple discrete model of the central pattern generator derived from the continuous model of Baxter et.al. [4]. The model of a single neuron is a simple generic qualitative "integrate and fire" model. We use the *same* model for each of the ten neurons, but specialize the generic model for some of the "special" neurons in the network. The electrical and synaptic interconnections are also modeled at a highly abstract qualitative level. There are no parameters that need instantiation in our model. The properties exhibited by the model are solely a consequence of the abstract model of neuron behavior and the positive and negative *interconnections*.

We will complement our informal presentation of the discrete model with a formal description in the Symbolic Analysis Laboratory (SAL) language [15]. SAL is a formal language for describing discrete state transition systems. We do not present a detailed introduction to the syntax and semantics of SAL here. Since SAL syntax uses inspiration from standard imperative and functional languages, readers unfamiliar with SAL can still possibly understand and appreciate the formal description. The full SAL model is available online [18].

4.1 Discrete Model of a Single Neuron

A generic neuron is a simple input/output automaton. It receives an input `i`, changes its internal state `level` in response to it, and optionally produces an output `o`. In our model, the input `i` can be one of three qualitative values: `pos`, `neg`,

```
N: NATURAL = 4;
LEVELS: TYPE = [0 .. N];
SIGS: TYPE = { pos, neg, zero };
NEURONS: TYPE = { B31, B35, B63, B34, B64, B4, B51, B52, B8, Z };
PHASES: TYPE = { protraction, retraction, termination };
GNEURONS(n: NEURONS): BOOLEAN = ( n /= B31 AND n /= B64 );
```

Fig. 2. SAL Global Declarations. `N` is a constant. `LEVELS, SIGS, NEURONS, PHASES` are types denoting finite sets. `GNEURONS` is a function that returns true when its argument is a generic neuron, and false when it is a specific neuron (B31, B64).

and `zero`, collectively referred to as `SIGS`. The value of `i` indicates whether the neuron receives a positive (`pos`), negative (`neg`), or no (`zero`) impulse (from its neighbors). The internal state of the neuron stores a value in the range $[0, \ldots, N]$ in a variable called `level`[1]. A value of 0 indicates that the neuron is at its resting potential, and N indicates that the neuron is at its highest membrane potential. The values in between represent abstractions of the concrete intermediate membrane potentials. The output `o` is a Boolean-valued variable. A value of `TRUE` indicates that the neuron emits an impulse, whereas `FALSE` indicates that it does not do so.

Figure 2 shows the declarations for the parameter N[1] and types used to describe the complete discrete CPG model. The type `NEURONS` denotes the set of all ten neurons, while the function `GNEURONS` identifies the eight that are generic. The type `PHASES` denotes the set of three phases.

The dynamics of single generic neurons are given by the following intuitive rules. The rules abstractly capture the "integrate and fire" behavior of neurons.

Integrate Positive Input Impulse (IPII): If the input impulse `i` is positive (`pos`) and the level `level` is less-than N, then the level `level` is incremented. The amount of increment is given by a parameter `sens`. Again, the properties of the model are robust to changes in the value of `sens`. We use `sens` $= 2$ in our experiments.

Integrate Negative Input Impulse (INII): If the input impulse `i` is negative (`neg`) and the level `level` is less-than N, then the level `level` is decremented. The amount of decrement is fixed to 2 units. The increments and decrements are always saturated so as to force the value of `level` to remain in the range $[0, \ldots, N]$.

Fire: If the level `level` is equal to N, then it is reset to $N - 2 * \texttt{sens}$ (which is 0 for the choices made above). Additionally, the output `o` is set to `True` whenever `level` $= N$ to indicate that the neuron fired.

If none of the conditions are applicable, then the state of the neuron remains unchanged. In particular, this means that we do not model decay of the

[1] We could use any positive value for N. We used the value 4 in our model, but the choice is really arbitrary and we could have used a different value, say 5 or 3: the properties of the final CPG model do *not* dependent on the exact value used.

```
generic[n: NEURONS]: MODULE = BEGIN
  INPUT i: SIGS
  OUTPUT o: BOOLEAN, level: LEVELS
  LOCAL sens: [1 .. 2], pir: BOOLEAN
  INITIALIZATION
     pir = FALSE; level = 0
  DEFINITION
     sens = IF (n = Z) THEN 1 ELSE 2 ENDIF;
     o = (level = N)
  TRANSITION
     [
     FIRE: level = N AND GNEURONS(n) AND NOT(pir) -->
             level' = N - 2*sens
     []
     IPII: level < N AND i' = pos -->
             level' = IF (level > N - sens) THEN N ELSE level + sens ENDIF
     []
     INII: (NOT(GNEURONS(n)) OR level < N) AND i' = neg -->
             level' = IF (level > 1) THEN level - 2 ELSE 0 ENDIF
     []
     SPIR: n = B52 AND level = 0 AND i = neg AND i' = neg -->
             pir' = TRUE; level' = N
     []
     ELSE -->
     ]
END;
```

Fig. 3. SAL Model of a Generic Neuron. The model is parameterized by the name n of the neuron. The model is specialized for (a) neurons that exhibit a plateau-like potential (identified by NOT(GNEURONS(n)) in the code above), (b) the neuron $B52$ that exhibits postinhibitory rebound (PIR). The ELSE clause says that when all the above conditions fail, do not change the state of the system. The neuron model responds to the value of i in the "next" step (i') because the module computing i (`aplysia_wiring` described later) already introduces a one-step delay.

membrane potential (`level`) with time. This is because in the time intervals of interest, decay does not play an important role in determining the overall behavior of the system. Also note that the model of a single neuron is *deterministic*. The conditions for the three cases above are mutually exclusive and given an input `i` and level `level`, the next state of the neuron is deterministically specified.

The formal description of the above neuron model is given in SAL in Figure 3. Since some of the neurons exhibit behavior that can not be entirely captured by an "integrate and fire" model, the generic model has a couple of specializations. The first one is for neurons that exhibit a plateau-like potential. In the model of the CPG, neurons B31 and B64 fall in this category [10, 16]. They are modeled to behave just like the other neurons, except that they do not fire; that is, the

```
aplysia_neurons: MODULE =
   (WITH INPUT ins: ARRAY NEURONS OF SIGS
    WITH OUTPUT outs: ARRAY NEURONS OF BOOLEAN
    WITH OUTPUT levels: ARRAY NEURONS OF LEVELS
      (|| (n: NEURONS): (RENAME o to outs[n], i to ins[n],
                                level to levels[n] IN generic[n])));
```

Fig. 4. SAL model of the collection of the ten neurons in the CPG. For example, `ins[B63]`, `outs[B63]`, and `levels[B63]` will now denote the input signal for B63, the output generated by B63 and the internal level of B63.

membrane potential (`level`) is not reset (to 0) when it reaches its highest value (N). As we shall later discuss, this distinction is important for the CPG to exhibit the observed behavior.

The second specialization is for neuron B52 that exhibits postinhibitory rebound (PIR). PIR is defined as membrane depolarization (activation) occurring at the offset of a hyperpolarizing stimulus. The mechanism for PIR is not well understood. In our abstract model, we modeled it in the form of a special transition (labeled `SPIR` in Figure 3) that depolarizes B52 when it is at its resting potential and it receives an inhibitory pulse. Again, this special behavior of B52 is important for ensuring termination of the ingestion and egestion neural programs. Neither of these special behaviors can be exhibited by a simple "integrate and fire" model [12].

4.2 Modeling a Collection of Neurons

We get a model of each of the ten neurons in the CPG network (enumerated in Figure 2) by instantiating the model of a generic neuron described above ten times. Thus, we get a collection of ten identical neurons. The few subtle distinctions between these ten neurons have already been captured in the description of the generic neuron described above.

Figure 4 describes the SAL code for generating models of the ten neurons. The ten instantiations of the generic neuron are synchronously composed (|| is the synchronous composition operator). The input `i`, output `o`, and the level `level` variables are renamed to avoid a naming conflict.

The result is a collection of ten neurons, but there is no interconnection between them yet. In the next section, we will model the interconnections.

4.3 Modeling the Interconnects

As mentioned before, there are three types of connections between the neurons in our model: excitatory synaptic connection, inhibitory synaptic connection, and electrical connection.

Excitatory Synaptic Connection. This is a *directed* connection between a source and a target neuron. In this connection, the source neuron produces an excitatory postsynaptic potential (EPSP) in the target neuron. This connection is modeled as follows: whenever the source neuron fires (that is, generates a `TRUE` signal on its output port `o`), the target neuron receives a `pos` input on its input port; otherwise, it receives only a `zero` input.

```
epsp(x: BOOLEAN): SIGS = IF x THEN pos ELSE zero ENDIF;
```

Inhibitory Synaptic Connection. This is a *directed* connection between a source and a target neuron. In this connection, the source neuron produces an inhibitory postsynaptic potential (IPSP) in the target neuron. This connection is modeled as follows: whenever the source neuron fires (that is, generates a `TRUE` signal on its output port `o`), the target neuron receives a `neg` input on its input port; otherwise, it receives only a `zero` input.

```
ipsp(x: BOOLEAN): SIGS = IF x THEN neg ELSE zero ENDIF;
```

Electrical Coupling. Electrical connections are undirected, that is, they effect both neurons that are coupled by an electrical connection. Electrical coupling indicates that there can be flow of current between the two coupled neurons that is proportional to the *difference* in the membrane potentials of the two cells. Since the membrane potentials are abstracted to qualitative levels (`level`), we model electrical coupling using the difference in the levels of the two neurons. Specifically, if the levels of electrically-coupled neurons, say A and B, are $\texttt{level}_A$ and $\texttt{level}_B$ respectively, then
(a) neuron A receives a `pos` input and neuron B receives a `neg` input if $\texttt{level}_B > \texttt{level}_A$;
(b) neuron A receives a `neg` input and neuron B receives a `pos` input if $\texttt{level}_B < \texttt{level}_A$; and
(c) both neurons receive a `zero` input if $\texttt{level}_B = \texttt{level}_A$.

```
diff(x1: LEVELS, x2: LEVELS): SIGS =
IF (x1>x2) THEN pos ELSIF (x1<x2) THEN neg ELSE zero ENDIF;
```

Accumulating Effects from Multiple Connections. The above three cases describe the input each neuron receives from each of its neighboring neurons. Each neuron now has to accumulate all its input signals and map the result to one value: `pos`, `neg`, or `zero`, that it will use as its actual input signal.

The accumulation process is modeled using the following simple rules: If `pp` denotes the total number of `pos` inputs and `nn` denotes the total number of `neg` inputs received by the neuron, then
(1) if $\texttt{pp} > \texttt{nn}$, then the result is a `pos` signal.

```
aplysia_wiring: MODULE =
BEGIN
 INPUT b63i: SIGS, outs: ARRAY NEURONS OF BOOLEAN
 INPUT levels: ARRAY NEURONS OF LEVELS
 OUTPUT ins: ARRAY NEURONS OF SIGS
INITIALIZATION ins = [ [i: NEURONS] zero ]
TRANSITION
[ TRUE -->
ins' = [ [n:NEURONS] LET
 ec:[[NEURONS,NEURONS]->SIGS]=LAMBDA(a,b:NEURONS):diff(levels[a],levels[b]),
 ep:[NEURONS->SIGS] = LAMBDA(x:NEURONS): epsp(outs[x]),
 ip:[NEURONS->SIGS] = LAMBDA(x:NEURONS): ipsp(outs[x]) IN
 IF (n=B31) THEN integrate31(ip(B64),ep(B34),ep(B63),ep(B35),
                             ep(B4),ec(B63,B31),ec(B35,B31))
 ELSIF (n=B34) THEN integrate(ep(B63),ip(B64),zero,...,zero)
 ELSIF (n=B63) THEN integrate(ip(B64),ec(B31,B63),ep(B34),b63i,...)
 ELSIF (n=B35) THEN integrate(ip(B64),ec(B31,B35),ip(B52),...)
 ELSIF (n=B64) THEN integrate64(ep(Z),ip(B52),ip(B34),ip(B63),
                                ip(B4),ep(B51),ec(B51,B64))
 ELSIF (n=B4) THEN integrate(ip(B52),ep(B35),ec(B51,B4),ep(B64),...)
 ELSIF (n=B52) THEN integrate(ep(B35),ip(B64),ip(B51),ip(B4),...)
 ELSIF (n=B8) THEN integrate8(ep(B51),ip(B52),ip(B63),ip(B4),ep(B34))
 ELSIF (n=Z) THEN integrate(ep(B63),...) ]
 ELSE integrate51(ip(B35),ip(B52),ip(B4),ec(B64,B51),ec(B4,B51))
]
END;
```

Fig. 5. SAL Model of the Interconnections of the ten neurons in the CPG. Neurons B31, B64, B8 and B51 have their own special integrate functions. Missing arguments, indicated by ..., are all zero.

(2) if `pp` $<$ `nn`, then the result is a `neg` signal.
(3) Otherwise, the result is a `zero` signal.

```
integrate(x1: SIGS, x2: SIGS, ..., x7: SIGS): SIGS =
   LET pp:[0..7] = count(pos, x1,x2,x3,x4,x5,x6,x7),
       nn:[0..7] = count(neg, x1,x2,x3,x4,x5,x6,x7) IN
     IF (pp > nn) THEN pos
     ELSIF (nn > pp) THEN neg
     ELSE zero ENDIF;
```

The Wiring Diagram. Figure 5 contains the final wiring diagram for the ten neurons. For each neuron, the `integrate` function described above is used to collect all inputs for that neuron. Depending on the type of connection, each input is obtained using either `epsp`, `ipsp` or the `diff` function.

Some of the neurons, namely B31, B64, B51 and B8, use a specialization of the `integrate` function described above. (Hence the names of the function used in

```
observer: MODULE =
BEGIN
  OUTPUT b63i: SIGS, phase: PHASES
  INPUT levels: ARRAY NEURONS OF LEVELS
INITIALIZATION b63i = pos
DEFINITION
  phase = IF (b63i=pos OR levels[B31]>=N-1) THEN protraction
          ELSIF (levels[B64]=N OR levels[B4]=N) THEN retraction
          ELSE termination ENDIF
TRANSITION
  [ levels[B35]=N --> b63i' = zero
  [] ELSE --> ]
END;
```

Fig. 6. SAL Model of the Observer: It generates the input trigger and observes phase changes.

Figure 5 for these neurons are different.) The specialization captures *preference* of some neurons to some excitatory or inhibitory inputs; that is, strength of some connections is stronger than others. The `integrate` function treats all connections equally. If we use the `integrate` function to accumulate inputs *for all* the neurons, then the resulting model's behavior does not match the observed behavior. We therefore specialized `integrate` for neurons B31, B64, B8 and B51 as follows: B31 requires inhibition from B64 to see a negative input, B64 gives high priority to signals from Z and B52, B8 gives lower priority to the inhibitory signal from B63, and B51 gives higher priority to its synaptic connections and lower to its electrical coupling with B4. These priorities can be captured by weighting the inputs and doing a weighted sum in the `integrate` function. These specializations are discussed further in Section 6.

4.4 Exciting the System and Observing the Phases

We wish to study the behavior of the system elicited by a brief depolarization of B63. We add a separate component to our model to inject an abstract depolarizing current pulse to B63. Figure 6 contains the SAL description of this additional "observer" module. The module begins by sending a `pos` input to B63, but as soon as B35 reaches its firing threshold (value N), the external input to B63 is reset to zero. This simulates the effect of giving B63 a brief depolarizing current pulse.

The protraction and retraction phases are characterized by activity in, respectively, the P_Group and the R_Group neurons (Figure 1). The "observer" module observes these phase changes: depending on the levels of B31 (a P_Group neuron) and B64 (a R_Group neuron), it decides whether the phase is protraction, retraction, or termination.

4.5 The Complete Model: Putting it All Together

We have completely described the models of all components of the model, namely the ten neurons, the connections between the neurons, and the observer. The final complete model is simply a synchronous composition of the three components, which is described in SAL as:

```
aplysia: MODULE = aplysia_wiring || aplysia_neurons || observer;
```

Synchronous composition means that at each abstract time step, each of the components (the ten neurons, the wiring, and the observer) simultaneously take a transition.

We analyzed the above discrete abstract model using model checking tools (described in detail in Section 5) and found that the CPG generates an egestion-like behavior. As suggested in the literature [4, 13], external modulatory influences, mostly through the neurotransmitter dopamine, can alter the CPG and cause it to exhibit an ingestion-like behavior. Following the approach of [4], we modeled the effect of dopamine by changing the strength of some of the interconnections. Specifically, we decreased the excitability of B34 and B4. We achieved this by specializing the `integrate` functions for B34 and B4. This resulted in a different model that is revealed to have ingestion-like behavior by model checking.

5 Analysis

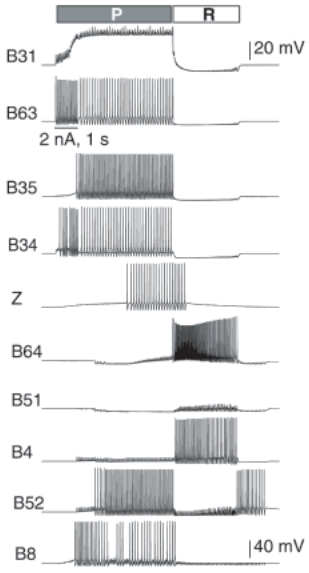
The discrete model described above is appealing for its simplicity of description and the lack of any requirement for hundreds of parameters. Moreover, we can use model checking to systematically explore the system for desired behaviors.

As a basic sanity check, we can verify that if there is no input trigger to B63, then all neurons in the system indeed remain in their resting potentials. Excitation of B63 by an external pulse initiates either the ingestion, or the egestion buccal motor program. Figure 7 shows plots generated by the continuous model of [4]. This bursting pattern is typical of the egestion BMP. We can capture the salient features of the patterns in Figure 7 in the form of Linear Temporal Logic (LTL) formulas. The LTL properties can then be model checked against the developed abstract qualitative model of Section 4. This gives us a way to validate our model against (experimental) observations.

We classify the properties into different groups depending on their pertinent phase.

5.1 The Protraction Phase

The protraction phase is characterized by activity in B63, B31, and B35. We first make sure that the input pulse to B63 is modeled correctly by checking that it is initially present (`pos`), and then terminated (`zero`) before the system reaches the retraction phase. Property `p0` in Figure 8 formally states this fact in LTL.



The plots show the activity in each of the ten neurons of the modeled CPG when B63 is excited by a short pulse. The excitation causes activity in the neurons in the protraction group (P-group) neurons (B31, B35, B63) . In the egestion mode, B34 and B8 are active in the protraction phase. Towards the end of the protraction phase, Z starts firing that in turn causes activity in B64. This immediately causes the P-group neurons to switch "off", while the retraction group (R-group) neurons (B4, B64) remain active. In this retraction phase, B8 is inactive, which indicates that this pattern corresponds to egestion. Activity in B52 signals an end of the retraction phase. B51 shows little activity throughout the two phases. In the ingestion mode (not shown here), the plots of B34, B8, B4, and B51 are different. Essentially, during ingestion, B34 remains mostly inactive, B8 shows no activity during protraction, but is active during retraction, B4 shows reduced activity, and B51 shows higher activity.

Fig. 7. [Figure 5 from [4]] Simulating the continuous model of the CPG [4] generates patterns characteristic of egestion. These plots are formalized as LTL properties in Figures 8, 9,10, and 11.

An important feature in Figure 7 is that B31 (eventually) reaches a plateau and stays there all through the protraction phase and until the start of retraction. This is captured in Figure 8 Property `p1`.

The neurons B63 and B35 show periodic firings in the protraction phase. Property `p2` partly encodes this fact as follows: at all points until two steps before retraction starts, it holds that B35 eventually fires. The same can be stated and verified for B63. Note that non-plateau neurons, such as B35 and B63, reset upon firing, and hence Property `p2` says that B35 *repeatedly* (and not necessarily periodically) fires. In the egestion scenario, the same is also true for B34 and Property `p3` captures this. Also during egestion, B8 fires during the protraction phase (Property `p4`).

We successfully verified all the above properties on our discrete model. Property `p3` and Property `p4` are verified for the egestion scenario, but, as expected, they fail when the model is modified for ingestion. Instead, for the ingestion model, the following property is verified, which states that B8 remains inactive all through the protraction phase.

```
p5: THEOREM aplysia ⊢ G(phase=protraction => levels[B8] < N);
```

```
p0: THEOREM
    aplysia ⊢ U(b63i = pos, U(b63i = zero, phase = retraction));
p1: THEOREM
    aplysia ⊢ F(levels[B31]=N AND U(levels[B31] >= N-1, phase=retraction));
p2: THEOREM
    aplysia ⊢ U( F(levels[B35] = N), X(X(phase = retraction)) );
p3: THEOREM
    aplysia ⊢ U( F(levels[B34] = N), X(X(levels[B64] = N)) );
p4: THEOREM
    aplysia ⊢ F(phase = protraction AND levels[B8] = N);
```

Fig. 8. Temporal Properties for the Protraction Phase in SAL. $F(A)$ means that A holds eventually, $U(A, B)$ means that eventually B holds and until then A holds, and $X(A)$ means that A holds in the next time step.

```
t1: THEOREM
    aplysia ⊢ F( levels[Z] = N );
t2: THEOREM
    aplysia ⊢ F( b63i = zero AND levels[B64] = N );
t3: THEOREM
    aplysia ⊢ F( levels[B64] = N AND G( levels[B63] < N ) );
t4: THEOREM
    aplysia ⊢ F( phase = retraction AND levels[B64] = N );
```

Fig. 9. Temporal Properties for the Transition Phase in SAL. The notation $G(A)$ means that A holds always from that instance onwards.

5.2 Transitioning from Protraction to Retraction

Figure 9 contains LTL properties pertaining to the switching off of the protraction phase and transitioning into the retraction phase. The main events in the transition from protraction to retraction are:
(a) the hypothetical neuron Z becomes active (Property `t1`),
(b) this causes the depolarization (activation) of B64, (Property `t2`),
(c) this, in turn, simultaneously causes the hyperpolarization (deactivation) of the protraction group neurons, such as B63, (Property `t3`), and
(d) eventually the retraction phase neurons, such as B64, are active (Property `t4`).
Property `t3` also states that B63 remains deactivated ever after. The same can also be said for the other protraction group neurons.

The properties above verify that the system eventually transitions from the protraction to the retraction phase. These properties hold true for the egestion, as well as the ingestion, model.

5.3 The Retraction Phase

Figure 10 contains LTL properties pertaining to the retraction phase. During retraction, B64 remains active (Property `r0`). The protraction phase neurons

```
r0: THEOREM
    aplysia ⊢ G( phase = retraction => levels[B64] = N );
r1: THEOREM
    aplysia ⊢ G( phase = retraction => levels[B35] < N );
r2: THEOREM
    aplysia ⊢ G( phase = retraction => levels[B34] < N );
r3: THEOREM
    aplysia ⊢ W( F(levels[B4] = N), X(X(phase = termination)) );
r4: THEOREM
    aplysia ⊢ G( levels[B51] < N );
r5: THEOREM
    aplysia ⊢ G( phase /= protraction => levels[B8] < N );
```

Fig. 10. Temporal Properties for the Retraction Phase in SAL. The LTL operator W is the "weak until" operator. $W(A, B)$ says that A continues to hold until B becomes true. Unlike $U(A, B)$, here B may never become true.

remain inactive at all instances during the retraction phase. Property `r1` states this for neuron B35, but the property can be stated and verified for the other P-group neurons as well. The same is also true for the neuron B34 (Property `r2`). Furthermore, the neuron B4 repeatedly fires during retraction. Again, this fact is partly encoded in LTL as follows: (Property `r3`) at all instances until two steps before the termination phase starts, it is true that the neuron B4 eventually fires.

The next two properties are specific to egestion. The neuron B51 is not part of the egestion behavior, but participates only during ingestion. Property `r4` states that B51 always remains inactive. Finally, Property `r5` states that the radula-closer motor neuron, B8, is inactive during the retraction phase (in fact, at any non-protraction state).

All the properties described above, except `r4`, are verified to be valid for egestion. Properties `r0, r1` and `r2` remain valid even when the model is specialized to the ingestion case. However, the remaining properties, Properties `r3, r4` and `r5` are, as expected, invalid for ingestion. While the plots in Figure 7 suggest that Property `r4` should be valid for egestion, it is not so for our model. We discuss this further in Section 6.

5.4 Termination

Figure 11 contains LTL properties pertaining to the termination of the ingestion/egestion Buccal Motor Program. As is evident from Figure 7, the termination phase is characterized by the hyperpolarization (deactivation) of all the protraction group neurons and the retraction group neurons. Property `e1` says that when the system enters the termination phase, eventually B64 and B4 return to their resting levels. Note that B64 and B4 may not be at their resting potential when the termination phase begins, and hence the eventuality operator (F) is important. The same fact can be stated for the P-group neurons, B31,

```
e1: THEOREM
    aplysia ⊢ G(phase=termination => F(G(levels[B64]=0 AND levels[B4]=0)));
e2: THEOREM
    aplysia ⊢ G(phase=termination => G(levels[B31]<N AND levels[B63]<N));
e3: THEOREM
    aplysia ⊢ U(phase=protraction, W(phase=retraction, phase=termination));
```

Fig. 11. Temporal Properties for the Termination Phase in SAL.

B35, and B63. Property `e2` states that B31 and B63 neurons always remain inactive during the termination phase. Since we are stating that the neurons are inactive (levels $< N$), and not asking for levels to be 0, we do not need the eventuality operator (F) here.

Finally, we state one of the most important properties of the CPG network: the protraction phase is followed by the retraction phase, which in turn is (optionally) followed by the termination phase – exactly in that order. This is stated in Property `e3`, which basically says that the three phases occur sequentially.

The three termination phase properties are model checked and verified to hold for both the egestion and the ingestion model.

6 Results and Discussion

The main observations from the study of the discrete model are described below. Most of the observations made here are similar to those obtained by working with the continuous model based on using Hodgkin-Huxley-type models for the neurons and several hundreds of parameters. Thus, by just looking at the interconnections at a fairly abstract level, it is still possible to build and analyze useful and interesting models that help in refining our understanding of the way a neuron network works.

Specialized neurons. B31 and B64 are different from the other neurons because they exhibit a plateau-like potential [10, 16]. This difference is important to sustain the activity in the protraction (B31) and retraction (B64) phases. If B31 and B64 are modeled in the same way as the other neurons, then the modified model does not exhibit the sustained activity of B31, B35, and B63 during protraction and that of B64 and B4 during retraction.

Electrical couplings. The effect of electrical coupling needs to be necessarily asymmetric to enable the model to have the desired behavior. In a symmetric scenario, if two neurons A and B are electrically coupled and if $\texttt{level}(A) > \texttt{level}(B)$, then $\texttt{level}(A)$ decrements as $\texttt{level}(B)$ increments. In an asymmetric scenario, we are allowed to have, say, $\texttt{level}(B)$ increase while $\texttt{level}(A)$ remains unchanged. In the model, using a symmetric effect on the B31 - B35 coupling and the B31 - B63 coupling would cause the protraction phase to prematurely end. Similarly, B64 - B51 coupling can cause the retraction phase to terminate

earlier if the effect is symmetric. We note that electrical coupling is asymmetric in the continuous model [4].

Robust protraction phase. The positive feedback loops between the protraction-phase neurons (B31, B35, B63, B34) lead to a very robust protraction phase, that is, once the system settles into the protraction phase (characterized by periodic firing of these neurons), it is not "easy" to drive the system out of that phase. In fact, we had to make the inhibition of the protraction-phase neurons by B64 a very "strong" signal to really cause the protraction phase to terminate. The transition from protraction to the retraction phase is not very well understood and had led to the hypothetical neuron Z in the model [4]. Our observation here suggests that apart from the unknown component Z, there is another important aspect related to the strengths of the synaptic connections between the P-group and the R-group neurons that is required to ensure transition into retraction.

Retraction phase. Unlike the protraction-phase neurons that are connected in a strong positive feedback, the neurons active in the retraction phase do not form any positive feedback cycle. As a result, the retraction phase is not robust and it can be "easily" terminated. Again, its continued sustainability depends crucially on the intrinsic ability of B64 to maintain a plateau-like potential, *despite the negative (inhibitory) feedback from its inter-connections*. Note that in the speculative model used here (proposed in [4]), B64 gets only inhibitory inputs in the retraction phase of egestion.

Ingestion. The default model exhibits *egestion* behavior, that is, the radula motor neuron (B8) is active (radula is closed) during protraction and inactive (radula is open) during retraction. As suggested in the literature [4, 13], if the excitability of cells B34 and B4 is reduced (characteristic of dopaminergic modulation), then the situation changes and B8 is inactive (radula is open) during protraction and active (radula is closed) during retraction. This corresponds to the ingestion BMP. The explanation for this change is as follows: reduced excitability of B34 causes it to remain inactive during the protraction phase, and hence B8, which depended on B34 for excitatory pulse, remains inactive too. On the other hand, during retraction, the low excitability of B4 causes it to remain relatively inactive, which allows B51 to activate and cause B8 to activate as well. Note that the mode change is solely explained by the network and the connections and does not depend on the detailed physical modeling of the single neurons or their synaptic connections.

B51. In contrast to the prediction made by the continuous model of [4], B51 is activated during egestion in our model. However, it does not affect the activation pattern of B8. It is possible to reduce the responsiveness of B51 to positive signals and have it remain inactive during egestion. However, in that case, it also tends to remain inactive during ingestion: the reduced inhibition by B4 (during ingestion) is not enough to overcome the reduced responsive of B51 to positive signals. These results indicate that the unmodeled components – the

sensory input neurons and the motor neurons – may have a significant effect on firing of B51.

Specialization versus Robustness. One important feature of biological networks is that they are robust to minor changes. The system continues to have the same behavior (equivalently, satisfy the same set of properties) even when certain changes are made to it. The positive feedback in the P-group neurons that sustains the protraction phase is such an example. Most properties that describe the ingestion and egestion behaviors are, in fact, robust to minor changes in the discrete model proposed in this paper. This robustness is the reason why abstract discrete models are useful. However, some of the properties of our model are quite sensitive to the strengths of certain synaptic connections. For examples, the interconnections of B64, and to some extent those of B51, B8, and B31 (note that we had to use specialized accumulator functions for these neurons, see Figure 5), appear to influence the overall behavior. The specializations indicate the sensitive parts of the model. This suggests that we may have to refine our current understanding of the CPG.

Weighted Integrate Function. Biological data shows that certain connections between neurons are stronger than others. Our definition of the weighted integrate functions is chosen to reflect this fact. The exact integrate function is, however, not important and other qualitatively similar functions produce the same behavior.

Model Checking Effort. Though we have used our in-house model checking environment SAL for experiments, we could have used any other temporal logic model checker. We used the symbolic model checker, `sal-smc`, which uses Boolean Decision Diagrams (BDDs) to represent states. Each model checking run (one for each property) takes about 10 seconds. The total state space of the system is of the order of 4^{10}; however, the set of reachable states must be significantly less. Adding more non-determinism to the model increases the model checking effort.
We also note here that, for any reasonable choice for the parameters, N, increment, and decrement, the model satisfies the same set of LTL properties. Our specific choice, such as N=4, was a reasonable compromise between being small and giving enough qualitative states (N=0,1,2,3) to model details (of other kinds of neurons we foresee adding to the network.)

7 Conclusions

We presented an abstract discrete model of the central pattern generator responsible for generating the egestion and ingestion buccal motor programs during the feeding cycle in *Aplysia*. We formalized the neural activation patterns, observed during egestion and ingestion BMPs, using LTL formulas. We verified that our model satisfies the LTL formulas using a symbolic model checker. While many

properties are a direct consequence of the excitatory, inhibitory, and electrical connections between the various neurons, some of the properties – especially those related to transitioning from protraction phase to the retraction phase and finally to the termination phase – depend on the relative strengths of the various synaptic and electrical connections.

We plan to expand the model to include other missing elements from the feeding neural circuit, such as the sensory neurons in the cerebral ganglion and the cerebral-buccal interneurons. Simultaneously, we will need to expand the collection of LTL properties to capture more functions and behaviors.

A second direction for future work is generalizing the present model and making it nondeterministic. The present model is deterministic, except for the initiation of the termination phase. We can drop assumptions and make our model more nondeterministic, while still preserving its properties. For example, when a neuron receives both positive and negative inputs, it presently behaves as if it received no input – implicitly assuming that all signals are of "equal strength". Instead, we can drop this assumption and let the neuron nondeterministically behave as if it received a positive, negative, or zero input.

References

[1] L.F. Abbott. Single neuron dynamics: an introduction. In Ventriglia F., editor, *Neural Modeling and Neural Networks*, pages 57–78. Pergamon Press, 1994.

[2] D.A. Baxter, C.C. Canavier, and J.H. Byrne. Dynamical properties of excitable membranes. In Byrne and Roberts [3], pages 161–196.

[3] J. H. Byrne and J. L. Roberts, editors. *From Molecules to Networks: An Introduction to Cellular and Molecular Neuroscience*. Elsevier, 2004.

[4] E. Cataldo, J. H. Byrne, and D. A. Baxter. Computational model of a central pattern generator. In *Proc. Computational Methods in Systems Biology CMSB*, volume 4210 of *LNBI*, pages 242–256, 2006. Private communication.

[5] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez. Pathway logic: Symbolic analysis of biological signaling. In *Proc. Pacific Symposium on Biocomputing*, pages 400–412, 2002.

[6] C. Elliott and A. Susswein. Comparative neuroethology of feeding control in molluscs. *J. Exp. Biol.*, 205:877–896, 2002.

[7] R. Ghosh, A. Tiwari, and C. Tomlin. Automated symbolic reachability analysis with application to delta-notch signaling automata. In *Hybrid Systems: Computation and Control HSCC*, volume 2623 of *LNCS*, pages 233–248. Springer, 2003.

[8] R. Grosu, S. Mitra, P. Ye, E. Entcheva, I. V. Ramakrishnan, and S. A. Smolka. Learning cycle-linear hybrid automata for excitable cells. In *HSCC*, volume 4416 of *LNCS*, pages 245–258, 2007.

[9] A. V. M. Herz, T. Gollisch, C. K. Manchens, and D. Jaeger. Modeling single-neuron dynamics and computations: A balance of detail and abstraction. *Science*, 314, October 2006.

[10] I. Hurwitz, R. Goldstein, and A. Susswein. Compartmentalization of pattern-initiation and motor function in the B31 and B32 neurons of the buccal ganglia of *aplysia californica*. *J Neurophysiol*, 71:1514–1527, 1994.

[11] S. M. Iyengar, C. Talcott, R. Mozzachiodi, E. Cataldo, and D. A. Baxter. Executable symbolic models of neural processes. In *Network Tools and Applications in Biology NETTAB07*, 2007.

[12] E. M. Izhikevich. Which model to use for cortical spiking neurons? *IEEE Trans. on Neural Networks*, 15(5), 2004.

[13] E. A. Kabotyanski, D. A. Baxter, S. J. Cushman, and J. H. Byrne. Modulation of fictive feeding by Dopamine and Serotonin in *aplysia*. *J. Neurophysiol.*, 83:378–392, 2000.

[14] P. Lincoln and A. Tiwari. Symbolic systems biology: Hybrid modeling and analysis of biological networks. In R. Alur and G. Pappas, editors, *Hybrid Systems: Computation and Control HSCC*, volume 2993 of *LNCS*, pages 660–672. Springer, March 2004.

[15] The SAL intermediate language, 2003. Computer Science Laboratory, SRI International, Menlo Park, CA. `http://sal.csl.sri.com/`.

[16] A. J. Susswein and J. H. Byrne. Identification and characterization of neurons initiating patterned neural activity in the buccal ganglia of *aplysia*. *J Neurosci*, 8:2049–2061, 1988.

[17] A. J. Susswein, I. Hurwitz, R. Thorne, J. H. Byrne, and D. A. Baxter. Mechanisms underlying fictive feeding in aplysia: Coupling between a large neuron with plateau potentials activity and a spiking neuron. *J Neurophysiology*, 87(5):2307–2323, 2002.

[18] A. Tiwari. SAL model of *aplysia* central pattern generator, 2008. `http://www.csl.sri.com/~tiwari/html/cmsb08.html`.

[19] A. Tiwari, C. Talcott, M. Knapp, P. Lincoln, and K. Laderoute. Analyzing pathways using sat-based approaches. In *Proc. 2nd Intl. Conf. on Algebraic Biology, AB 2007*, volume 4545 of *LNCS*, pages 155–169. Springer, 2007.