Building a Spiking Neural Network Model of the Basal Ganglia on SpiNNaker

Basabdatta Sen-Bhattacharya, *Senior Member, IEEE*, Sebastian James, Oliver Rhodes, Indar Sugiarto, Andrew Rowley, Alan B. Stokes, Kevin Gurney, and Steve B. Furber, *Fellow, IEEE*

***Abstract*—We present a biologically inspired and scalable model of the basal ganglia (BG) simulated on the spiking neural network architecture (SpiNNaker) machine, a biologically inspired low-power hardware platform allowing parallel, asynchronous computing. Our BG model consists of six cell populations, where the neuro-computational unit is a conductance-based Izhikevich spiking neuron; the number of neurons in each population is proportional to that reported in anatomical literature. This model is treated as a single-channel of action-selection in the BG, and is scaled-up to three channels with lateral cross-channel connections. When tested with two competing inputs, this three-channel model demonstrates action-selection behavior. The SpiNNaker-based model is mapped exactly on to SpineML running on a conventional computer; both model responses show functional and qualitative similarity, thus validating the usability of SpiNNaker for simulating biologically plausible networks. Furthermore, the SpiNNaker-based model simulates in real time for time-steps $\geq$ 1 ms; power dissipated during model execution is $\approx$ 1.8 W.**

***Index Terms*—Action selection, basal ganglia, biologically-inspired neural network, Izhikevich neuron model, neuromorphic, SpineCreator, SpineML, SpiNNaker.**

Manuscript received July 25, 2017; revised November 15, 2017 and January 18, 2018; accepted January 20, 2018. Date of publication March 8, 2018; date of current version September 7, 2018. This work was supported in part by the U.K. Engineering and Physical Sciences Research Council (EPSRC) through the Universities of Southampton, Cambridge, and Sheffield, and industry partners, ARM Ltd., Silistix Ltd., and Thales under Grant EP/D07908X/1 and Grant EP/G015740/1, in part by the European Union (EU) ICT Flagship Human Brain Project through many university and industry partners across the EU and beyond under Grant FP7-604102, and in part by the European Research Council through the EU's Seventh Framework Programme (2007–2013) under ERC Grant 320689. The work of B. Sen-Bhattacharya was supported by the ERC Grant. The work of S. James and K. Gurney was supported in part by the EC FP7 Project NoTremor under Grant 610391, and in part by the EC H2020 Project Dreams4Cars under Grant 731593. The work of I. Sugiarto was supported by the EPSRC under Grant EP/L000563/1. *(Corresponding author: Basabdatta Sen-Bhattacharya.)*

B. Sen-Bhattacharya, O. Rhodes, A. Rowley, A. B. Stokes, and S. B. Furber are with the School of Computer Science, University of Manchester, Manchester M13 9PL, U.K. (e-mail: basab.sen.b@gmail.com).

S. James is with the Department of Automatic Control Systems Engineering, University of Sheffield, Sheffield S1 2LT, U.K.

I. Sugiarto was with the School of Computer Science, University of Manchester, Manchester M13 9PL, U.K. He is now with the Department of Electrical Engineering, Petra Christian University, Surabaya, Indonesia (e-mail: indi@petra.ac.id).

K. Gurney is with the Department of Psychology, University of Sheffield, Sheffield S1 2LT, U.K.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCDS.2018.2797426

I. Introduction

THE AIM of this paper is to build a biologically inspired, scalable, spiking neural network model of the Basal Ganglia (BG) on the spiking neural network architecture (SpiNNaker) machine [1]. The BG are a set of subcortical nuclei, which are evolutionarily very old and appear in all vertebrates, enabling them to make decisions and take subsequent actions [2]. The information on which the decision needs to be made, i.e., the environmental circumstance, constitutes the input to the BG, and is available via the thalamus and cortex. Output from the BG is the specific action that is decided upon, referred to as "action-selection" [3], and is relayed to the motor pathway for execution via the thalamus, cortex, and other subcortical structures. The objective of this paper is to build a computational framework that will provide a "basic building block" for further testing and development of automated decision-making tools in low-power, real-time hardware such as the SpiNNaker machine [4], [5].

The neurotransmitter dopamine lies at the heart of the decision-making/action-selection functions of the BG. Seminal modeling work by Gurney *et al.* [3], [6] introduced the concept of "selection-control" pathways in the BG, a deviation from the more common nomenclature of "direct–indirect" pathways associated with how dopamine controls and executes the action-selection mechanism. Subsequently, the model was also demonstrated as a computational tool to study brain disorders [7]–[9], as well as to form a conceptual understanding of the action-selection mechanism adopted by the BG and implemented in robots [10], [11].

The BG model presented in this paper is based on the selection-control BG model proposed by Gurney *et al.* [3], and its later extension by Humphries *et al.* [9]. This circuitry consists of four cell populations, namely: 1) the subthalamic nucleus (STN); 2) globus pallidus externa (GPe); 3) substantia nigra pars reticulata (SNr); and 4) the striatal medium spiny neurons (Str-MSN). In addition, we have included the striatal fast spiking interneurons (Str-FSI) along the lines presented in [12] (sans the gap junction connections). Furthermore, our BG network model comprises several recurrent connections that are based on literature reporting anatomical data. The basic computational unit in our model is implemented using Izhikevich's conductance-based spiking neurons, supported by SpiNNaker's underlying software toolchain sPyNNaker [13]. Our choice of the single spiking neuron model is inspired by a similar implementation in the BG models presented by Liu *et al.* [14] and Thibeault and Srinivasan [15]. Inputs to

the model are simulated with Poisson distributed spike trains generated separately for each simulation run, and are provided to the Str-MSN, Str-FSI, and STN cell populations. The response of the SNr cell population is the model output. At first, we built a macroscopic "channel"-like columnar model capturing neural information pathways in the BG, and parameterized this model to produce base firing rates as reported in [9] and [15]. Next, to emulate arbitration by the BG of parallel macroscopic information channels representing competing sources, the single-channel model is used as a basic building block to scale up to three channels. Our results demonstrate selection of a competing action by the three-channel BG model, and are in agreement with previous model-based research [9], [15].

To compare and validate the SpiNNaker-based model outputs with those obtained using a conventional computer, we have implemented the same BG circuit using SpineML [16], an XML-based format for the specification of networks of point-neuron models. To create the SpineML-based model, we made use of SpineCreator, a graphical editor that is designed to provide an easy-to-use and flexible interface for building and visualizing neuronal models [17], [18]. The model output dynamics show functional and qualitative similarity on both platforms, i.e., SpiNNaker and SpineML, indicating the SpiNNaker machine as a viable platform for implementing spiking neural networks. A stringent bootstrapped t-test [19] (see the Appendix) performed on the total number of spikes generated by each population over a period of 6 s shows that $p < 0.05$, implying statistically significant numerical differences between the model spike counts. This difference is due to the stochastic nature of the model inputs, replicating the numerical differences between data recorded from different animals for the same behavioral task, and is aligned with our expectations.

The underlying SpiNNaker architecture is designed to run in real time for time-steps $\geq$ 1 ms. However, we solve the Izhikevich neuron models with a time-step of 0.1 ms to ensure solution accuracy. Thus, all simulations of the BG model on SpiNNaker ran in 10 s real time for 1-s simulation time, i.e., slowed down by a factor of 10. That said, a performance analysis indicates that the model is guaranteed to execute in this time, which lends a reliability factor, e.g., for real-time implementations. In addition, the run time is unaffected by scaling up the model, i.e., 10-s model simulation time is guaranteed to execute in 100 s real time for both single- and three-channel models. The power dissipation during model execution, measured using equipment built in-house [20], is $\approx$ 0.8 and 1.8 W for the single- and three-channel model, respectively.

In Section II, we present the model design and implementation methods. In Section III, we present the model simulation methods and results. A comparison study with simulation of the same model on the SpineML platform running on a conventional computer is presented in Section IV-A; a performance analysis in terms of simulation time and power dissipation on SpiNNaker is presented in Section IV-B. We discuss the results and conclude this paper in Section V.

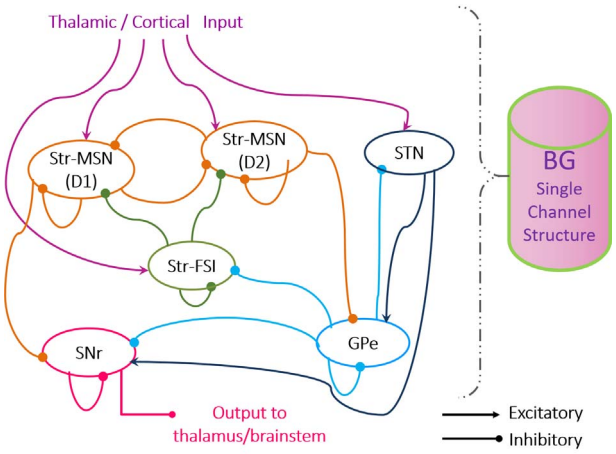

Fig. 1. Schematic of the single-channel BG model. An overview of the biological basis of the model layout is provided in Section II-A. A pool of 25 Poisson distributed spike trains provide input to the Str-MSN and Str-FSI populations, while a separate pool consisting of two Poisson spike trains provide input to the STN population. Model output is the average firing rate of all neurons in the SNr population.

II. Model Design and Implementation

A. Biological Background

The basic BG model circuitry simulated on SpiNNaker is shown in Fig. 1. The Striatum forms the main input structure of the BG and receives excitatory glutamatergic synapses from both cortex and thalamus. Studies on the BG cells of the rat brain [21] report that around 90%–95% of the cells of the Striatum are of the Str-MSN variety. The remaining 5%–10% of the cells constitute the interneurons of the Striatum. While there are three known varieties of interneurons, the predominant inhibitory influence on the Str-MSN is thought to be from the γ-aminobutyric acid (GABA)-ergic Str-FSI, which constitute around 2%–5% of the cells of the Striatum [22], [23]. In this paper, we model the Str-FSI population constituting around 3% of the cells of the Striatum.

A core feature of the BG is the modulation of population behavior by dopamine released by the substantia nigra pars compacta (SNc; not modeled here). The Str-MSNs receive major dopaminergic input from the SNc and are modulated selectively by two types of dopamine receptors, classified broadly as D1 and D2. The D1 receptors are known to facilitate N-methyl-D-aspartate (NMDA) and $GABA_A$ mediated synapses [24]–[26], while the D2 receptor types suppress the α-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid (AMPA) and $GABA_A$ mediated synapses [27], [28].

The Str-MSN cells that are modulated by the D1 receptors (Str-MSN-D1) make major inhibitory axonal projections to the SNr. The Str-MSN cells that are modulated by the D2 receptors (Str-MSN-D2) project to the GPe. The GPe cells project inhibitory efferents to the STN, SNr, Str-FSI [29], as well as on themselves [30]. Both Str-MSN-D1 and Str-MSN-D2 have recurrent inhibitory connections on themselves in addition to laterally inhibiting one another [28]. The Str-FSIs make feed-forward inhibitory synapses on both Str-MSN-D1 and Str-MSN-D2 [31], as well as within the population. We note that the STN are the only excitatory cell population in

TABLE I
TOTAL NUMBER OF NEURONS IN EACH POPULATION OF THE BG MODEL IS INFORMED BY ANATOMICAL STUDIES [21] AND SCALED DOWN PROPORTIONALLY TO FORM A COLUMNAR STRUCTURE, REPRESENTING A SINGLE-CHANNEL OF NEURAL INFORMATION FLOW IN THE BG, CONSISTING OF A TOTAL OF 2681 CELLS

Basal Ganglia population	Total number of neurons (reported)	Total number of neurons (model)
STR (Str-MSN + Str-FSI) (N_{str})	2790×10^3	2790
Str-MSN (N_{str}^{msn})	$90\% \times N_{str}$	2511
Str-MSN-D1/D2	$50\% \times N_{str}^{msn}$	1255
Str-FSI	$3\% \times N_{str}$	84
STN	13560	14
GPe	45960	46
SNr	26320	27

the BG; all other cells are inhibitory. The STN cells receive major excitatory inputs from the thalamus and cortex [32], and project excitatory efferents to the GPe and SNr populations. The SNr forms the output structure of the BG and projects inhibitory efferents to the ventral thalamus and the brainstem reticular formation. In addition, the SNr cells make inhibitory projections on other cells within the population [33].

Table I shows the total number of cells in each population of a rat BG as reported by Oorschot [21]. However, in this preliminary endeavor to model the BG circuit on SpiNNaker, our objective is to build a single-channel columnar architecture that will serve as the basic building block toward building multichannel BG models; the justification toward such an approach is to build a scalable framework. Toward this, we scale down the number of neurons in each population by a factor of 10^3. While there are a myriad chemical neurotransmissions in the BG, we have implemented only two types of synapses in this paper, viz., those mediated by the AMPA and $GABA_A$ neuro-receptors corresponding to glutamatergic (excitatory) and GABA-ergic (inhibitory) neurotransmitters. Further details on synaptic layout and parameterization of the network are mentioned in the following sections.

B. Single Neuron Models and Spiking Patterns

Each computational unit in the BG network is a conductance-based form of Izhikevich's [34] spiking neuron model implemented on the SpiNNaker software toolchain sPyNNaker [13], and defined in the following equations:

$$\frac{dv(t)}{dt} = 0.04v^2(t) + 5v(t) + 140 - u(t) + I_{\text{dc}} + I_{\text{syn}}(t) \quad (1)$$

$$\frac{du(t)}{dt} = a(bv(t) - u(t)) \quad (2)$$

$$\text{If } v(t) > 30 \text{ then } v(t) \leftarrow c;\ u(t) \leftarrow u(t) + d \quad (3)$$

where $a, b, c,$ and d are parameters that define the dynamic behavior of the model and can be tuned to obtain various spiking patterns as observed in biology; $v(t)$ is the membrane potential and $u(t)$ is a membrane recovery variable; I_{dc} is the DC bias current that is built into the model definition in sPyNNaker; and $I_{\text{syn}}(t)$ is the post-synaptic current corresponding to the synaptic processes mediated by the neuro-receptors

TABLE II
(a) BASE (REFERENCE) STATE PARAMETERS OF SINGLE NEURON MODELS IN EACH POPULATION OF THE BG MODEL DEFINED IN (1)–(3) [14], [15], [35]. READERS MAY NOTE THAT THE BASE VALUES FOR I_{dc} ARE SET DURING THE SIMULATION OF THE WHOLE NETWORK DISCUSSED IN SECTION III-A. (b) BASE STATE PARAMETER VALUES FOR SYNAPTIC AND DOPAMINERGIC MODULATION ATTRIBUTES DEFINED IN (4)–(9)

(a)

Basal Ganglia	a	b	c	d	v_{init} (mV)	u_{init}	I_{dc} (nA)
Str-MSN	0.02	0.2	-65	8	-80	-16	-30
Str-FSI	0.1	0.2	-65	8	-70	-14	-10
GPe	0.005	0.585	-65	4	-70	-40.95	2
STN	0.005	0.265	-65	2	-60	-15.9	5
SNr	0.005	0.32	-65	2	-70	-22.4	5

(b)

Neurotransmitter	Parameters	Values
AMPA	τ_{ampa}	6 ms
	$\bar{g}_{ampa}$	0.5
	E_{ampa}	0 mV
	$\phi_{dop}^{str-msn-d2}$	2.75
	$\phi_{dop}^{str-fsi}$	3.75
	ϕ_{dop}^{stn}	2
	ϵ_{ampa}^{d2}	0.2
GABA	τ_{gaba}	4 ms
	$\bar{g}_{gaba}$	0.5 of $\bar{g}_{ampa}$
	E_{gaba}	-80 (mV)
	$\phi_{dop}^{str-msn-d1}$	2.75
	$\phi_{dop}^{str-msn-d2}$	2.75
	$\epsilon_{gaba}^{d1,d2}$	0.073

syn ∈ {AMPA, GABA_A} and is defined in the following equations:

$$I_{\text{syn}}(t) = g_{\text{syn}}(t) \cdot \big(E_{\text{syn}} - v(t)\big) \quad (4)$$

$$g_{\text{syn}}(t) = g_{\text{syn}}(t_0) \cdot e^{-(t-t_0)/\tau_{\text{syn}}} \quad (5)$$

$$g_{\text{syn}}(t_0) = g_{\text{syn}}(t_0 - \Delta t) + n\bar{g}_{\text{syn}} \quad (6)$$

where E_{syn} and $g_{\text{syn}}(t)$ are the membrane reversal potential and membrane conductance, respectively, of the post-synaptic neuron; τ_{syn} is the decay time constant of the synapse; $g_{\text{syn}}(t_0)$ is the instantaneous conductance after the most recent afferent spike; n is the total number of spikes incident at the synapse in the time-step (Δt) before t_0 implemented on the ring buffer of sPyNNaker (see Section II-D); and $\bar{g}_{\text{syn}}$ is the conductance increment per afferent spike (the "synaptic weight").

Initially, each population of the BG is simulated on SpiNNaker with neither any interpopulation or intrapopulation connectivities nor any extrinsic model input ($I_{\text{syn}}(t) = 0$). The dc bias current I_{dc} corresponding to a population X forms part of the neuron definition on sPyNNaker, and is present as an intrinsic input stimulus to all neurons in X from the start of simulation time. In this state, a single simulation run of the BG model will generate spiking behavior from all neurons in the model, where each neuron is responding to I_{dc} only, and is otherwise acting independent of all other neurons in the model. Such a set-up allows us to specify a "base state" spiking pattern for each population of the model; specifically, we aim to emulate the spike patterns demonstrated by Thibeault and Srinivasan [15] (see Fig. 1) generated on a conventional computer. The spiking patterns generated on SpiNNaker are shown in Fig. 2. The total duration of simulation is 5 s, where I_{dc} is varied after every 1 s of simulation

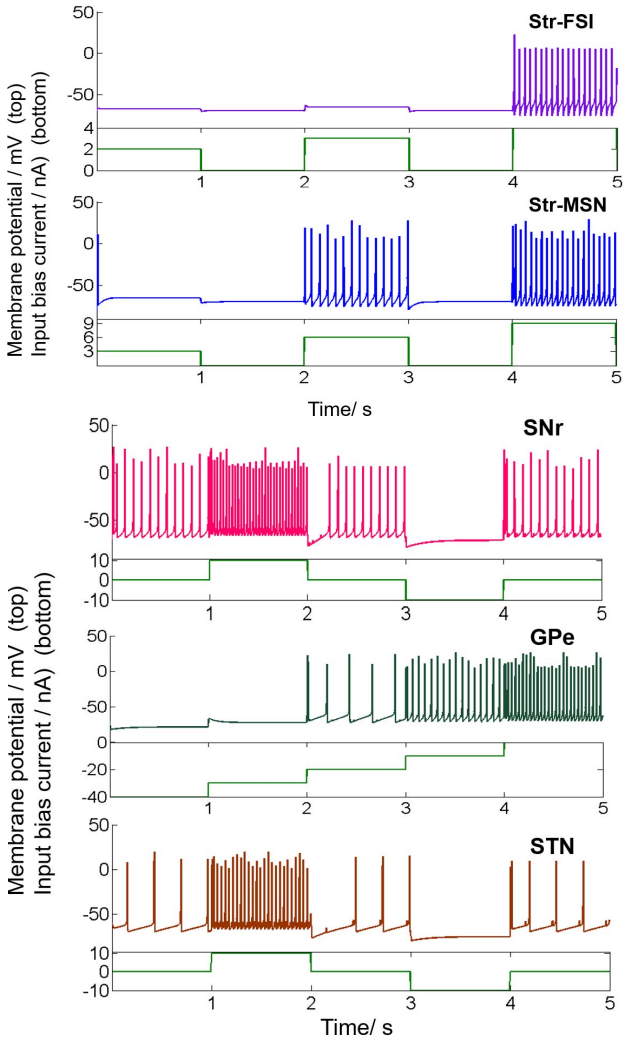

Fig. 2. Spiking patterns of the Izhikevich neurons in each population of the BG model when stimulated with I_{dc} varying every 1 s over the total simulation time of 5 s. There are no other stimuli in the circuit, and the neurons are acting independently with no intranetwork or internetwork connectivity. Each neuron is defined as in (1)–(3) with model parameters set as in Table II(a). The parameterization of SNr, GPe, STN, and Str-MSN cell populations was made with the objective to match, qualitatively, those in [15, Fig. 1]. We did not find any prior research demonstrating the spiking patterns for Str-FSI cells and parameterized these populations as fast spiking (FS) [35].

time; this demonstrates the spike response characteristics of each population in terms of increasing or decreasing frequency corresponding to changes in I_{dc}. The base (reference) state parameters of the single neuron models (with the exception of I_{dc}) for each cell population in this paper are mentioned in Table II(a), and are informed by those in [14] and [15] [readers may note that the base values for I_{dc} mentioned in Table II(a) are set during simulation and testing of the BG network as discussed in Section III-A].

C. Synaptic Layout and Dopaminergic Modulation

In a recent review [32], all cell populations in the BG are reported as expressing both AMPA and NMDA neuroreceptors corresponding to glutamatergic neurotransmitters, and both GABA_A and GABA_B neuroreceptors corresponding to GABA neurotransmitters. However, the GABA_B neuroreceptor mediated synapses are modulatory in nature, and they do not participate actively in the synaptic transmission process. Furthermore, the NMDA neuro-receptor-based synapse is a function of the membrane voltage. For simplicity, we ignore the NMDA- and GABA_B-based neuro-transmission in this paper, and consider synapses mediated by the AMPA and GABA_A neuro-receptors only.

The reversal potential for the GABA_A neuro-receptor mediated synapses depends on the flow of Cl^-, and its value relative to the resting potential of the cell affects the nature of the post-synaptic membrane potential [36]–[38]. Here, we assume the case where a GABA_A mediated synapse would generate an inhibitory post-synaptic potential; thus we set the value to -80 mV [15]. The base parameter values of τ_{syn}, and E_{syn} of the post-synaptic membrane corresponding to syn $\in$ {AMPA, GABA_A} neuro-receptor mediated synapses [defined in (4)–(6)] are as in [9] and listed in Table II(b).

A literature survey indicates extensive dopaminergic modulation of synaptic transmission in all cell populations of the BG [23], [32], [39]–[45]. For simplicity, we have constrained the dopaminergic modulation in our BG model to only a few synaptic pathways as listed below. The mathematical implementation of dopamine modulation is applied to the peak membrane conductance corresponding to a synapse, and is informed by prior works [9], [46].

1) *Weakening of the AMPA Mediated Synapses:* The D2 receptors primarily target the AMPA neuro-receptor mediated synapses, weakening their impact by around 20% [24]. The D1 receptors primarily facilitate the NMDA mediated synapses while the AMPA mediated synapses are left unaffected. The modulation of the AMPA mediated excitatory afferents from extrinsic sources (thalamus/cortex) to the Str-MSN-D2, Str-FSI, and STN are implemented using the following equation:

$$\bar{g}^{\bar{P}}_{\text{ampa}-d2} = \bar{g}_{\text{ampa}} \cdot \left(1 - \epsilon^{d2}_{\text{ampa}} \cdot \phi^{\bar{P}}_{\text{dop}}\right) \tag{7}$$

where $\bar{P} \in$ {Str-MSN, Str-FSI, STN} represents the afferent populations receiving extrinsic inputs; $\epsilon^{d2}_{\text{ampa}}$ is the modulation co-efficient and is set as 0.2 to emulate the 20% modulation of the AMPA-based synapses [24]; and $0 < \phi^{\bar{P}}_{\text{dop}} < 5$ is the level of dopamine affecting the afferent synapse to the population $\bar{P}$. Thus, for a maximum value of $\phi^{\bar{P}}_{\text{dop}}$, $\bar{g}^{\bar{P}}_{\text{ampa}-d2} = 0$; conversely, for lack of dopaminergic modulation, i.e., $\phi^{\bar{P}}_{\text{dop}} = 0$, the maximal value of $\bar{g}^{\bar{P}}_{\text{ampa}-d2}$ is $\bar{g}_{\text{ampa}}$.

2) *Modulation of GABA_A Mediated Synapses:* The GABA-ergic inhibition of the GPe by the Str-MSN-D2 population is weakened by the D2 receptors [44], while the GABA-ergic inhibition of SNr by the Str-MSN-D1 population is facilitated by the D1 receptors [25]; it is, however, unclear whether such facilitation is via presynaptic or post-synaptic receptors. We have implemented both these modulatory pathways as follows:

$$\bar{g}^{\bar{P}}_{\text{gaba}-d1} = \bar{g}_{\text{gaba}} \cdot \left(1 - \epsilon^{d2}_{\text{gaba}} \cdot \phi^{\bar{P}}_{\text{dop}}\right) \tag{8}$$

$$\bar{g}^{\bar{P}}_{\text{gaba}-d2} = \bar{g}_{\text{gaba}} \cdot \left(1 + \epsilon^{d1}_{\text{gaba}} \cdot \phi^{\bar{P}}_{\text{dop}}\right). \tag{9}$$

TABLE III
SYNAPTIC CONNECTIVITY PARAMETERS CORRESPONDING TO BASE STATE OF THE SINGLE-CHANNEL BG MODEL. (a) AFFERENTS FROM THE CORTEX. EFFERENTS OF THE (b) STRIATUM, (c) FS INTERNEURONS, (d) GPE, (e) SNR, AND (f) STN

(a)

Synaptic connections	Synaptic Weight ($\bar{g}_{syn}$ μS)		Probability (p_{conn})	Delay (d_{conn} ms)
Ctx $\rightarrow$ Str-MSN-D1	$(\bar{g}_{ampa})$	0.5	0.15	9 – 12
Ctx $\rightarrow$ Str-MSN-D2	$(\bar{g}^{str-msn-d2}_{ampa-d2})$	0.225	0.15	
Ctx $\rightarrow$ FSI	$(\bar{g}^{str-fsi}_{ampa-d2})$	0.125	0.15	
Ctx $\rightarrow$ STN	$(\bar{g}^{stn}_{ampa-d2})$	0.3	0.2	

(b)

Synaptic connections	Synaptic Weight ($\bar{g}_{syn}$ μS)		Probability (p_{conn})	Delay (d_{conn} ms)
Str-MSN-D1 $\rightarrow$ SNr	$(\bar{g}^{str-msn-d1}_{gaba-d1})$	0.3	0.15	5 – 7
Str-MSN-D2 $\rightarrow$ GPe	$(\bar{g}^{str-msn-d2}_{gaba-d2})$	0.2	0.15	
Str-MSN-D1 $\rightarrow$ Str-MSN-D2 Str-MSN-D1 $\rightarrow$ Str-MSN-D1 Str-MSN-D2 $\rightarrow$ Str-MSN-D1 Str-MSN-D2 $\rightarrow$ Str-MSN-D2	$(\frac{1}{2.55} \times \bar{g}_{gaba})$	0.0982	0.1	2 – 3

(c)

Synaptic connections	Synaptic Weight ($\bar{g}_{syn}$ μS)		Probability (p_{conn})	Delay (d_{conn} ms)
Str-FSI $\rightarrow$ Str-MSN-D1 Str-FSI $\rightarrow$ Str-MSN-D2 Str-FSI $\rightarrow$ Str-FSI	$(\frac{1}{2.55} \times \bar{g}_{gaba})$	0.0982	0.1	2 – 3

(d)

Synaptic connections	Synaptic Weight ($\bar{g}_{syn}$ μS)		Probability (p_{conn})	Delay (d_{conn} ms)
GPe $\rightarrow$ STN	$(\frac{1}{1.75} \times \bar{g}_{gaba})$	0.1429		5 – 7
GPe $\rightarrow$ SNr			0.25	5 – 7
GPe $\rightarrow$ GPe				2 – 3
GPe $\rightarrow$ Str-FSI			0.05	5 – 7

(e)

Synaptic connections	Synaptic Weight ($\bar{g}_{syn}$ μS)		Probability (p_{conn})	Delay (d_{conn} ms)
SNr $\rightarrow$ SNr	$(\frac{1}{1.75} \times \bar{g}_{gaba})$	0.1429	0.25	2 – 3

(f)

Synaptic connections	Synaptic Weight ($\bar{g}_{syn}$ μS)		Probability (p_{conn})	Delay (d_{conn} ms)
STN $\rightarrow$ GPe STN $\rightarrow$ SNr	$(\frac{\bar{g}^{stn}_{ampa-d2}}{6})$	0.05	0.5	5 – 7 (intra-channel) 9 – 12 (cross-channel)

For simplicity in this paper, we constrain the dopaminergic modulation variability of GABA-ergic synapses to the MSN population only, and assume dopaminergic modulation of inhibitory afferents of both GPe and SNr from Str-MSN to be mediated by presynaptic D2 and D1 receptors, respectively. Thus, in (8), $\bar{P} \in \{\text{Str}-\text{MSN}-D2\}$, while in (9), $\bar{P} \in \{\text{Str}-\text{MSN}-D1\}$. Also, the co-efficient of dopaminergic modulation of GABA_A in both pathways is set to a base value of 7.3%, i.e., $\epsilon^{d1,d2}_{\text{gaba}} = 0.073$.

3) *Modulation of AMPA Efferents of the STN:* The STN sends out diffused excitatory projections to the GPe and SNr populations of the same channel as well as of neighboring channels (see Section II-D). Once again for simplicity, we have assumed presynaptic dopaminergic modulation of these AMPA mediated efferents, implemented using (7).

In addition to the above, the dopaminergic weakening of GABA_A mediated synapses is modeled by reducing the membrane conductance to a fraction of $\bar{g}_{\text{gaba}}$.

The base values of all the dopaminergic modulatory parameters are mentioned in Table II(b), while the values for the modulated conductance, $\bar{g}_{\text{syn}}$, are mentioned in Table III. These values are based on [9], although the final values are set by a "trial and error" approach during model simulation so as to obtain the target firing rate (see Section III-A). Readers may note that we did not vary the dopamine levels (Φ, ϵ) for this paper, and all results are generated with the base parameter values as mentioned in Table II(b). Thus, the dopamine modulation parameters in this model serve to set its operating region.

D. Overview of SpiNNaker and Its Handling of Synapse

SpiNNaker (Spiking Neural Network Architecture) is a system-on-chip consisting of very-low-power ARM968 processors. The on-chip communication architecture and protocols are biologically inspired, allowing asynchronous (event-based), parallel processing of synaptic data during neural network simulations on the ARM processors (referred to as "cores"). Each chip has 18 cores, of which around 15–16 are available for neural computation; the remaining cores are used for system management on the chip. Each core has 64-kB (data) tightly coupled memory (DTCM: analogous to a "cache" on a conventional computer, i.e., for quick data access during neural computation) where the neuron and synapse data pertaining to that core is stored to be accessed during

neural computation. In addition, each chip has a 128-MB synchronous dynamic random access memory (SDRAM) that is shared by all the cores on the chip for storing simulation data. For details of current state-of-the-art in SpiNNaker development, we refer the reader to a recent topical review [47]. The model used in this paper is implemented on a single 48-chip SpiNNaker board (please refer to Section II-D for details).

The SpiNNaker software toolchain, sPyNNaker [13], provides an implementation of PyNN [48], which is used as a standard interface for all neural simulations on SpiNNaker [PyNN is a python-based library bespoke to building spiking neural network models, and runs on other simulators (e.g., NEST) besides SpiNNaker]. Two essential criteria that form the backbone of the sPyNNaker application interface are: the neuron model, which we have discussed in Section II-B; and the synaptic connections in the network, which we describe in the following text.

The synaptic connectivity in the model is implemented via sPyNNaker using the function "FixedProbabilityConnector." Each connection between two populations consists of three attributes.

1) The probability of the synaptic connection $p_{\text{conn}} \in (0, 1)$, which is a normalized representation of the total "fan-in" from the presynaptic population to the post-synaptic population.
2) The delay of the synaptic connection d_{conn}, representing the latency of a presynaptic cell spike in reaching the post-synaptic cell.
3) The synaptic (connectivity) weight that scales the synaptic decay exponential, and is the membrane conductance increment per spike of the post-synaptic neuron $\bar{g}_{\text{syn}}$ in this paper [defined in (6)].

All of the above-mentioned synaptic attributes corresponding to a certain projection are stored as a 32-bit "synaptic data word"; the first 16 bits consist of the synaptic weight, while the next 16 bits are distributed thus: 4 bits for synaptic delay; 1-bit for synapse nature (i.e., excitatory/inhibitory); 8 bits for neuron index (therefore capped to 256 neurons per core); and 3 bits are unused. This is shown in the Appendix [Fig. 10(b)].

E. Mapping the BG Model to SpineML

A neuronal model that has been specified in SpineML consists of individual XML files, which define the behavior of model "components" viz., neuron bodies, post-synapses, and weight-updates. In addition, there are separate XML files that define how the components are built into a network of neuronal populations that are connected with "projections," where each projection consists of one weight-update and one post-synapse component. A set of "experiment files" define how the model should be executed; each experiment contains a specification of the inputs for the network, the data that should be logged from the simulation and any experiment-specific network lesions or parameter modifications that should be made. SpineML is thus a *declarative* format for specifying a network model.

In order to execute a SpineML model, it is necessary to use a SpineML back-end, which parses the SpineML input files and generates executable code for the model. We used SpineML_2_BRAHMS [49] that generates code suitable for execution on a general purpose CPU and is the canonical back-end for SpineML. The single-channel BG model on SpiNNaker in Fig. 1 is mapped to SpineML and shown in Fig. 3. Inspection of the figures indicates that the network connectivity is the same, although there is an important difference in solving the synapse models. Fig. 3 shows a typical SpineML model in which "spike events" are transmitted along projections via a weight-update component (a mechanism to implement fan-in from multiple presynaptic neurons) and then to the post-synapse component of the projection whose conductivity is incremented by the synaptic weight $\bar{g}_{\text{syn}}$. This $\bar{g}_{\text{syn}}$ is a parameter of the post-synapse component, which means that a spike afferent from population A and a spike afferent from a different population B increment the conductivity at the post synapse by the same amount. In contrast, in SpiNNaker, each spike is transmitted in a data packet *which encodes* $\bar{g}$ (see the ring-buffer implementation in Fig. 10), meaning that the spike from population A could increment the post synaptic conductivity by a different amount than the spike from population B. Thus, there can be a difference between SpineML and SpiNNaker networks which have apparently been arranged in an identical manner. It is possible to create a SpineML network which faithfully reproduces the behavior of the SpiNNaker network, but this leads to a more complex, unwieldy network (see the SpineML model *bgbsb*1_*impt* in the repository referenced below and compare with *bgbsb*1, which is shown in Fig. 3). The more complex, and more faithfully SpiNNaker-like SpineML model produced results that were not statistically different from those discussed in Section IV for the more natural SpineML models.

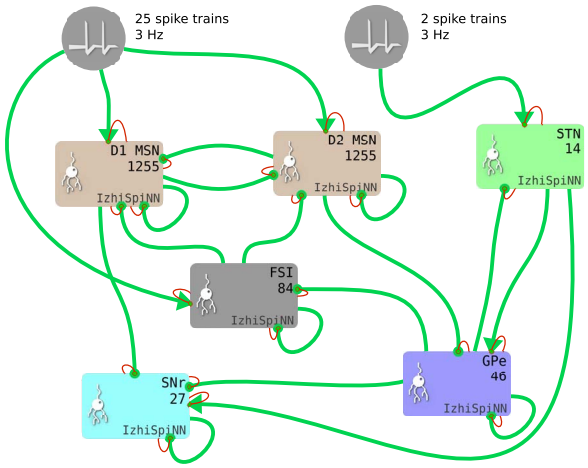

Fig. 3. Single-channel BG network, as exported from SpineCreator. Rectangular boxes are neural populations. The population name, number of elements, and SpineML component are shown in the box. Gray circles represent Poisson spike train sources. Green arrows are projections with element to element connectivities that are parameterized as in the SpiNNaker-based model and reported in Tables I–III. Projections with arrow heads are excitatory, those with circles for heads are inhibitory. The thinner, red lines which connect populations to arrowheads connect the membrane voltage variable $v(t)$ [defined in (1)] in the efferent population to the synapse component in each projection (on a one to one basis), allowing the synaptic current to be computed.

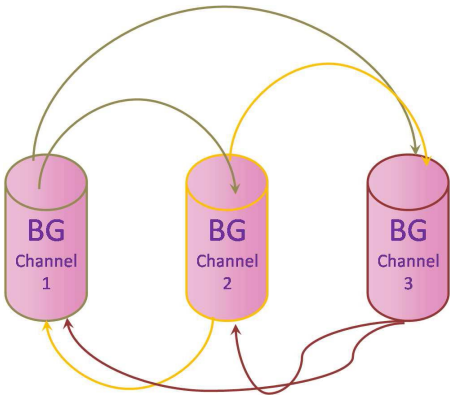

Fig. 4. Schematic of a scaled up model of the BG where three channels are interconnected laterally. The STN population of each channel projects to GPe and SNr populations of every other channel. The connectivity parameters are mentioned in Table III(f), and referred to as "cross-channel." The model demonstrates action-selection when presented with competing inputs (see Section III-B).

The SpineML model and associated results are available publicly at https://github.com/ABRG-Models/GPR-BSB/.

F. Scaled-Up BG Model With Three Channels

The BG has multiple parallel pathways that serve different parts of the cortex [50], as well as segregated voluntary and automatic behavior pathways [51]. Furthermore, focussed inhibition and surround excitation are concepts that were proposed by Mink [2]. We have scaled up our BG model to consist of three channels, where each channel is the single-channel model of Fig. 1. Thus, the total number of neurons in this three-channel BG model is 8043. The STN population of each channel sends out excitatory efferents to the SNr and GPe populations of the other two channels. These cross-channel connections produce the desired surround effect on the neighboring channels by each channel Ψ, thus indirectly emphasizing the focal inhibition within Ψ. The delay parameter of the cross-channel efferents of the STN is higher than that of the intra-channel pathway [see Table III(f)]. The connectivity parameters are mentioned in Table III. The method of initiating competing sources and incorporating action-selection in the model is discussed in Section III.

III. Simulation Methods on SpiNNaker and Results

A. Base State Dynamics of the Single-Channel Model

Data collected from BG of awake resting rats suggest firing rates of: STN at around 10–12 Hz; GPe at $\approx$ 30 Hz; SNr at somewhere in the range 25–30 Hz, and generally less than that of GPe [9]. Furthermore, the firing rate of the Str-MSN cells are $\approx$ 3 Hz while that of the Str-FSI are $\lesssim$ 10 Hz. We tuned the parameters in Tables II(b) and III to emulate these firing rates, which will define the base firing rates of the model populations.

1) Simulation Methods: To mimic extrinsic input to the BG, a total of 25 Poisson distributed spike sources project to the Str-MSN and Str-FSI populations, while two Poisson sources feed the STN population. The Poisson number for all the spike sources is maintained at 3 Hz. The total duration of simulation is 10 s. The Poisson sources are applied stochastically sometime between 500 and 700 ms from the start of simulation, and for a total duration of 9.2 s. The individual neuron model equations on sPyNNaker are solved using a second order Runge Kutta solver with a time-step 0.1 ms to achieve solution accuracy.

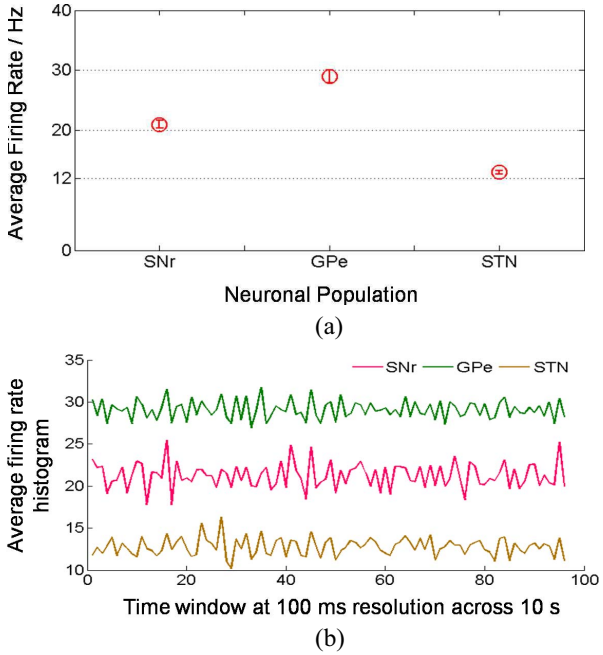

Fig. 5. (a) Average firing rates and (b) average firing rate histogram of SNr, GPe, and STN populations with time bin widths of 100 ms, computed over a simulation duration of 10 s, demonstrate similar ranges of mean firing rate as reported in [9] and [15].

The average firing rate of each population over r trials is derived using the following equation:

$$S_R = \frac{\sum_{r=1}^{R} S_m^r}{t_{\text{sim}} \cdot N_m \cdot R} \tag{10}$$

where S_m^r is the total number of spikes fired for each trial r by all N_m neurons in the population $M \in \{\text{SNr, GPe, STN}\}$, and over the total simulation duration of $t_{\text{sim}} = 10$ s; $R = 10$ is the total number of trial simulation runs. The spike count histogram is derived by averaging the spike count for each 100 ms bin of S_R.

2) Results: The average firing rates for the STN, GPe, and SNr populations are shown in Fig. 5(a), while the spike count histogram with 100-ms bins is shown in Fig. 5(b). The firing rate for: STN lies within the range 10–12 Hz; GPe lies within the range 30–32 Hz; and SNr lies within the range 20–25 Hz. Thus, our results show a good similarity with those reported in [9] (see Fig. 2) and [15] (see Fig. 6). We now treat this model as the single channel of decision-making and action-selection in the BG.

B. Simulating Action-Selection on SpiNNaker

Next, we aim to simulate action-selection using the three-channel BG model presented in Section II-F.

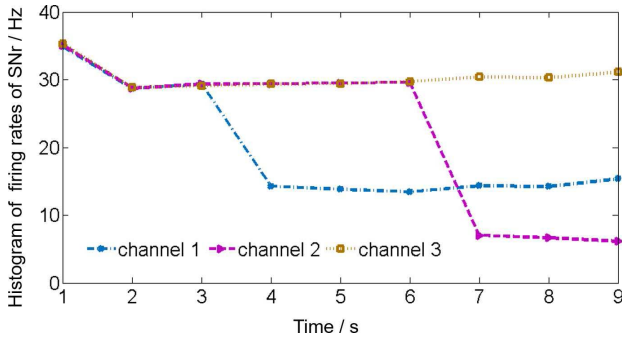

Fig. 6. Average firing rate histogram, with a bin width of 1 s, of SNr populations in the (blue dashed-dotted line) first, (pink dashed line) second, and (yellow dotted line) third channels of the BG model shown in Fig. 4. The model response demonstrates action-selection in the circuit. The third channel (yellow) does not receive any competing input and is therefore the neutral channel.

1) Simulation Methods: Similar to the single channel model, Poisson input at 3 Hz is provided to all the three channels from around 500–700 ms and for a duration of 9.2 s. Total simulation time is 10 s at a resolution of 0.1 ms. At $\approx t_1 = 3$ s from the start of simulation, the first channel receives a request for being "selected," which is simulated by providing Poisson spike trains at 15 Hz, drawn from two separate pools consisting of 25 and 2 Poisson spike sources, and provided to the Striatum and STN populations, respectively, in the channel. At $\approx t_2 = 6$ s, the second channel receives a request for being selected, which is simulated by 25-Hz Poisson spike trains provided to all the input pathway cells of the channel. Both 15- and 25-Hz spike train inputs are present until ≈ 9.9 s from start of simulation. Thus, all Poisson sources are withdrawn at ≈ 100 ms before the end of simulation; after this time and until the end of simulation, all channels respond to I_{dc} only. The outputs of the model are the SNr firing rates of all three channels. The results are averaged over ten trials. The average firing rate time histogram with time bin widths of 1 s is shown in Fig. 6. All dopamine levels were kept at base values indicated in Table II(b).

2) Results: Fig. 6 demonstrates action selection in the three-channel model simulated on the SpiNNaker machine. When a channel receives a Poisson input with a higher frequency, the focal inhibition of SNr within the channel increases. At the same time, the STN cells of that channel provide increased excitatory projections to the SNr cells of the neighboring channels, thus producing an "off-center" effect, and the firing rate of the SNr population in the competing channel drops relative to the other channels. We did not set a specific firing rate threshold to demonstrate selection of action, and rather let the inherent model dynamics take control. Thus, all parameters in each channel of the model are the same as those of the single-channel model.

IV. Comparison With SpineML and Performance Analysis

The model implemented on SpineML is intended to be topologically and parametrically identical to the one running on

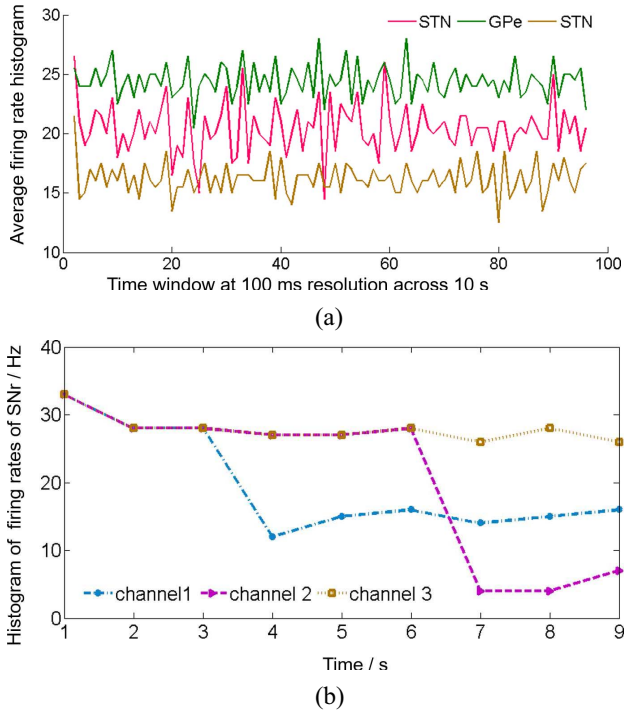

Fig. 7. (a) Average firing rate histogram of the SNr, GPe, and STN populations in the single-channel BG model simulated on SpineML_2_BRAHMS, with time bin widths of 100 ms across 10-s simulation time. The results demonstrate qualitative similarity with SpiNNaker-based simulation shown in Fig. 5. (b) Average firing rate histogram, with a bin width of 1 s, of SNr populations in the (blue dashed-dotted line) first, (pink dashed line) second, and (yellow dotted line) third channels of the BG model. Thus, the three-channel model response demonstrates action-selection; also, firing rates of the selected channels are similar to those on SpiNNaker shown in Fig. 6.

SpiNNaker. All parameters and network connectivities are configured the same in both systems. Fig. 7(a) shows the firing rate histogram of the SNr, STN, and GPe populations of the single-channel BG circuit simulated using the SpineML model. The figure demonstrates similar mean spiking rates for the STN and SNr as on the SpiNNaker-based model shown in Fig. 5; the firing rate lower bound of GPe cells are lower than that on SpiNNaker. The three-channel BG circuit on SpineML also demonstrates action selection behavior when simulated with exactly the same parameter and input attributes. This is shown in Fig. 7(b). The results demonstrate qualitative and functional similarity between models simulated on the two independent platforms.

A. Statistical Comparison of the Single-Channel Models

Although we did not expect the numerical results of simulations to be identical, we tested whether the two implementations would generate statistically equivalent results. Table IV gives the results of statistical tests on the spike counts in each population of the single-channel model obtained from 30 repetitions of 10-s simulations run on both SpiNNaker and SpineML_2_BRAHMS. The standard error of the mean number of spikes was computed by the bootstrap method with 256 resamples. The difference of the means is also given, along with a bootstrapped estimate of the difference of the means

TABLE IV
(A) MEAN SPIKE COUNT. (B) SPIKE COUNT DIFFERENCES BETWEEN SINGLE-CHANNEL BG MODEL SIMULATED ON SPINNAKER AND SPINEML. NUMBERS IN BRACKETS ARE BOOTSTRAP ESTIMATES OF STANDARD ERROR IN THE RESPECTIVE MEASURES. (C) ASL: A BOOTSTRAPPED EQUALITY OF MEANS TEST. ASL < 0.05, INDICATED IN BOLD, IMPLY MEANS ARE SIGNIFICANTLY DIFFERENT

	(A) Mean spike count		(B) Spike count difference	(C) ASL
BG population	SpiNNaker	SpineML	SpiNNaker - SpineML	SpiNNaker vs SpineML
Str-MSN-D1	3127 (4)	2869 (82)	258 (113)	**0.016**
Str-MSN-D2	0.33 (0.13)	0.5 (0.17)	-0.17 (0.2)	0.22
Str-FSI	1.73 (0.38)	1.47 (0.31)	0.27 (0.49)	0.28
STN	1072 (6.2)	1158 (1.7)	-86 (5.7)	**<0.0001**
GPe	8207 (6.9)	8518 (4.3)	-311 (7.6)	**<0.0001**
SNr	3510 (10.3)	3705 (4.1)	-195 (11.6)	**<0.0001**

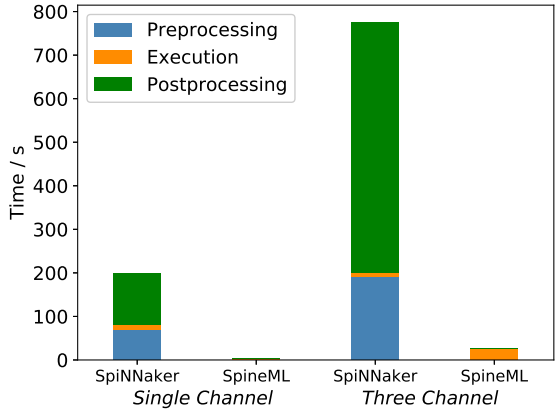

Fig. 8. Performance analysis of single- and three-channel BG models on SpiNNaker and SpineML for 1-s simulation time.

from 256 resamples. Finally a bootstrapped test of the difference of the mean number of spikes was applied with 10 000 resamples. The achieved significance level (ASL) is a measure of "the probability that the means are indistinguishable" (see the Appendix). The results indicate that whilst the spike counts are similar, none of the results can be said to be statistically equivalent according to this stringent test that the spike counts be indistinguishable in all populations.

B. Performance Analysis

The PyNN script describing the BG model is mapped and executed on the SpiNNaker machine by the sPyNNaker software toolchain [13], which itself runs on a host machine, in three stages: 1) preprocessing; 2) execution; and 3) postprocessing. *Preprocessing* involves translation of the PyNN-defined network into a form suitable for the SpiNNaker machine, and includes partitioning, routing, and generation and loading of data structures. In the context of performance testing, *execution* is defined as the time taken, once all data has been loaded, to run the simulation on the SpiNNaker machine. *Postprocessing* refers to extraction of resultant data, generated by executing the model, from the SpiNNaker machine to the host machine. Both uploading/extraction of data to/from the SpiNNaker machine is currently done via Ethernet [4].

The single-channel BG model consists of 2.68×10^3 neurons and $\approx 0.68 \times 10^6$ synapses (estimated from projection probabilities). While each processor within a SpiNNaker chip is capable of simulating an upper limit of 256 neurons (discussed in Section II-D), memory requirements of the neuron model and synaptic connectivity for certain applications may cause this number to be reduced. In this paper, sPyNNaker maps the single-channel BG model on to 32 cores distributed across two SpiNNaker chips, residing on a single 48-chip SpiNNaker board. In case of the three-channel model, the total number of neurons and synapses are 8.043×10^3 and $\approx 2.05 \times 10^6$, respectively, and the model network is mapped by sPyNNaker on to 96 cores, distributed across seven SpiNNaker chips.

Preprocessing is done on a 4-core 8-GB RAM desktop host machine, and takes 70.5 s for the single-channel BG model (three-channel: 191.0 s). The SpiNNaker hardware is designed to execute neuronal models in real time at a resolution ≥ 1 ms. Both single- and three-channel BG networks are configured to simulate with a solver time-step of 0.1 ms in order to maintain solution accuracy. Due to this constraint, 1 s of model simulation time is executed in 10 s real ("wall clock") time. However, both single- and three-channel models are guaranteed to execute within this 10 s. On execution completion, a further 119.1 s is required to extract output data for the single-channel model (three-channel: 574.7 s), giving an average total simulation time of 199.8 s for the single-channel model (three-channel: 776.0 s). The timing data recorded from the SpiNNaker execution of both single- and three-channel models is shown in Fig. 8. Timing values are averaged across ten repeated runs; the standard deviations across the ten samples of each model were less than 1.3 s, 4 ms, and 2.7 s for preprocessing, execution, and postprocessing, respectively.

The above-mentioned data for SpiNNaker-based model simulation is now compared to that using SpineML and executing on a 4-core 8-GB RAM desktop host machine, extracting and saving data, and "logs" (postprocessing). The results are also shown in Fig. 8, and indicate that the single-channel model simulated on SpineML performs preprocessing in 0.036 s, while 1 s of model simulation time is executed in 3.5 s real time. For the three-channel model, preprocessing time is 0.1 s, and the execution time equivalent of 1-s simulation time is increased significantly to 26.7 s real time. The postprocessing time is insignificant in both cases and ≈ 0. Clearly, the time of execution increases with scaling up of the model, and emphasizes the advantage of SpiNNaker-based computation for larger models over conventional computers.

1) Power Consumption on SpiNNaker: In a recent work, we used in-house Arduino-based power measurement equipment to measure power directly from a 48-node SpiNNaker board during model execution (the reader may refer to [20] for details). The main draw-back of this previous set-up was the coarse resolution (8.9 ms) of recording power from the SpiNNaker board. In this paper, we have used an enhanced (Raspberry-pi-based) version of this equipment, allowing a resolution of up to 0.6 ms with cleaner recording, i.e., without noise/glitches. Thus, the sampling rate of recording the

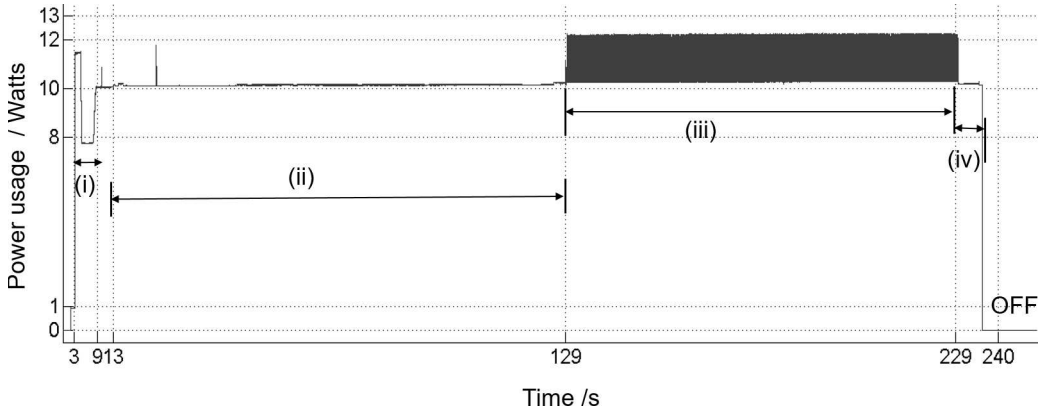

Fig. 9. Power consumptions of the three-channel model using an in-house Raspberry-pi-based measurement system connected to the SpiNNaker board (see [20] for details). The duration of recording the power can be broken down into four regions: 1) booting the machine; 2) preprocessing of data; 3) model execution; and 4) postprocessing (i.e., data extraction); the delay of around 4 s after booting the machine is inserted for clarity. The peak-to-peak power in region 3 is 1800 mW. The measurement sampling interval is 0.6 ms in real time. This is shorter than the time-step of model simulation (1 ms real time) in order to ensure that we do not lose data due to circuit delays between the Raspberry pi and the SpiNNaker board.

power is higher than 1 ms, the time-step of model simulation, and minimizes the potential for data loss due to delays during communication with the SpiNNaker board via Ethernet. This paper shows that the single-channel model execution uses $\approx$ 800 mW, while the three-channel model execution consumes $\approx$ 1.8 W shown in Fig. 9. The figure also confirms that the model execution time is not affected by scaling up to three channels, and is consistent at 100 s real time corresponding to a simulation time of 10 s. As power consumed during preprocessing and postprocessing are negligible compared to that during model execution, we kept the postprocessing time to a minimum; preprocessing times are handled by sPyNNaker and is not accessible to the user.

V. Conclusion

We have presented a biologically plausible and scalable model of the BG circuit, designed to run on the SpiNNaker machine—a biologically inspired architecture built with low-power ARM processors, allowing inherent asynchronous, parallel computation, and in real time for time-steps $\geq$ 1 ms. A single neuro-computational unit in our BG model is simulated with a conductance-based Izhikevich neuron model, facilitated by the underlying SpiNNaker software toolchain, sPyNNaker, which in turn is based on PyNN, a python-based neural network application interface. A columnar structure of the BG circuitry is first parameterized on SpiNNaker to set the base firing rates for all model cell populations, informed by existing literature. This forms the basic building block for a scalable framework, and is thought to be a single-channel for action-selection in the BG. To simulate action-selection by competing inputs, we scaled up the model to consist of three channels, and tested with two competing inputs in the presence of background noisy stimulus. Our results show that an input stimulus that is larger than the others is always the "winner," indicated by a relative drop in the firing rate of the SNr population (representing the BG model output) in the competing channel. The reduced firing rate of the inhibitory SNr population implies a reduced inhibition of the thalamic/brainstem cells, which are known to be the recipients of the BG output. This in turn means that the "action" that is solicited by a relatively larger ("competing") input is now "decided" by the BG circuit to be "selected, and acted upon," indicated by disinhibition of the target outputs. We have tested our model with a competing input of 15 Hz in the presence of a background noisy input of 3 Hz. This is further confirmed by "selection" of a larger input of 25 Hz provided in the presence of both 15- and 3-Hz inputs. On both occasions, the largest input wins. It is worth mentioning here that dopamine neurotransmitter-receptor levels are fundamental to facilitating decision-making and action-selection by the BG. Here, we tuned the base parameters simulating neutral dopamine levels; studying model dynamics with varying levels of dopamine will be carried out in future works.

To verify our model results simulated on SpiNNaker, we mapped the model to SpineML, an XML-based platform representing model attributes as components, and executing the models with SpineML_2_BRAHMS, a bespoke simulator which converts the SpineML model into machine code and runs it on a conventional computer. We aimed for the BG model implementation on SpineML to have the exact same network topology and neuron attributes as the SpiNNaker version, and therefore retained all model connectivities and parameter values used in the latter. Model results on SpineML show qualitative similarity with those on SpiNNaker in terms of base firing rates of the single-channel BG model cell populations. Implementation of the three-channel model on SpineML, following the exact same implementation procedures as on SpiNNaker, demonstrates action-selection by a larger input. Overall, the functional and qualitative behavior of the models are in agreement. A difference of means test (see the Appendix) indicates statistically significant numerical difference between the two platforms. We speculate that such difference is due to the stochastic nature of the model inputs, and simulates the numerical differences in recorded data from different brains, even when they are in the same

state, or performing similar behavioral tasks. We believe that our comparative study will provide a basic framework for mapping SpiNNaker-based models to SpineML, as well as for performance benchmarking of SpiNNaker with conventional computers during neuronal simulation.

The main drawback of our model is the inability to implement parameters that are voltage dependent, and thus need updating during run-time. Thus, we were unable to implement the voltage dependent NMDA synapses, nor the gap-junction (resistive) connections in the Str-FSI populations. This is due to current computational constraints on SpiNNaker during run-time, and work is ongoing to provide such implementations in the future. Another drawback is the slow Ethernet-based data transfer rates between the host-machine and the SpiNNaker. This is indicated in the performance analysis where the post-processing (data extraction) times are observed to increase significantly with scaling up of the model. In comparison, the preprocessing (mapping high-level model description to simulator) and postprocessing times for both single- and three-channel models implemented on SpineML are negligible. Model execution on SpiNNaker for this paper is slowed down by a factor of 10 relative to real time. This is because, the underlying Izhikevich equations need to be computed with a time-step of 0.1 ms to achieve solution accuracy, while the inherent SpiNNaker design is for real time operation with time-steps ≥ 1 ms. Thus, 10 s of model simulation time on SpiNNaker runs in 100 s real time. However, this execution time is guaranteed, i.e., both single-channel and three-channel BG models execute in 100 s real time corresponding to 10-s simulation time—this consistency demonstrates the ability of SpiNNaker to scale network size without compromising on execution time. In contrast, although the single-channel model execution time on SpineML is lower than that on SpiNNaker ($\approx$ 3 s real time for 1-s simulation time), that for the three-channel model scales up significantly, and by an order of 10 (approximately). Continuing research on the BG model implementation on SpiNNaker is looking into further scaling up of the model, which will serve to test and challenge the SpiNNaker machine on its real-time computational capabilities.

Continuing development of an in-house equipment is looking into ways to measure power directly from a 48-node SpiNNaker board during model execution [20]. To measure power during execution of the BG model, we use a Raspberry-pi-based system (enhancement from the Arduino-based system described in [20]), allowing the recording of power at 0.6 ms (real time) resolution. This is $\approx$ half the sampling resolution at which the model is set to execute (1 ms real time). The single-channel model uses two SpiNNaker chips (32 cores) and dissipates $\approx$ 0.8 W; the three-channel model runs on seven SpiNNaker chips (96 cores) and dissipates $\approx$ 1.8 W; the corresponding energy costs are 80 and 180 J, respectively. In comparison, the thermal design power for the CPU (Core i7 2600, 3.4 GHz) used to simulate the three-channel model on SpineML_2_BRAHMS is 95 W; considering 10-s simulation time is executed in 267 s real time (see Section IV-B), the energy cost for running the model on SpineML_2_BRAHMS is 25.36 KJ, which is $\gg$ 180 J on SpiNNaker. That said, currently, uploading any neural-network model to the SpiNNaker hardware is dependent on a host-machine with a standard CPU running the software toolchain sPyNNaker. Thus, a fair comparison between the two platforms would have to consider the CPU energy costs on the host-machine for SpiNNaker-based simulations, which is expected to be in the order of kilojoules for the preprocessing times reported in Section IV-B. Furthermore, while future robotic applications on SpiNNaker may not need postprocessing of data, all neuro-scientific investigations using SpiNNaker would have to rely on the host-machine for postprocessing of simulation data. Suffice to say that implications for power usage on SpiNNaker will need further testing and validation by running larger models, and will be looked into as future research.

In conclusion, this paper demonstrates the SpiNNaker platform as capable of simulating biologically plausible decision-making and action-selection circuitry that executes in a parallel and asynchronous manner, and within guaranteed time-scales. Furthermore, the platform demonstrates the potential for simulating large scale models without compromising on execution times. In addition, prior research has shown the low energy requirements of the SpiNNaker machine (e.g., compared to NEST [52]). Not surprisingly, therefore, use of SpiNNaker has been proposed in several robotic applications [53], [54]. Autonomous intelligent decision-making is a key desirable attribute in robotic applications, which can benefit wide-ranging societal requirements. We believe this paper developing the BG model on SpiNNaker will strengthen endeavors to build intelligent decision-making machines.

APPENDIX

A. Ring Buffer—Implementing Synapse

Let us assume an example case where a neuron-X, residing in core-X of one SpiNNaker chip, initiates a spike transfer that is to be delivered to the post-synaptic neuron-Y, residing in core-Y of the same chip, at a delay $d_{\text{conn}} = 3$ ms, and with $p_{\text{conn}} = 1$, thus guaranteeing a connection. After a series of activities initiated by this spike event (the details of which can be found elsewhere [1] and are outside the scope of this report), the first 16 bits of the synaptic data representing $\bar{g}_{\text{syn}}$ is now fetched from the chip's SDRAM and placed in a ring buffer of core-Y to be used for the post-synaptic membrane current computation for neuron-Y.

The ring buffer is a right circular shift-register structure occupying 16 kB of DTCM of each core, and is the basic algorithm that defines the post-synaptic behavior in an afferent neuron population. A depiction of the ring buffer is shown in Fig. 10(a). Each neuron in a core will have two rows in the ring buffer prebooked and at its disposal—one corresponding to an excitatory projection, and another corresponding to an inhibitory projection. Furthermore, each row consists of sixteen "slots," and each slot consists of 16 bits. To access the synaptic weight in the ring buffer at the appropriate delay slot, there is a right circular shift "pointer," which forms the reference for the specific slot in which the 16-bit synaptic weight is placed.

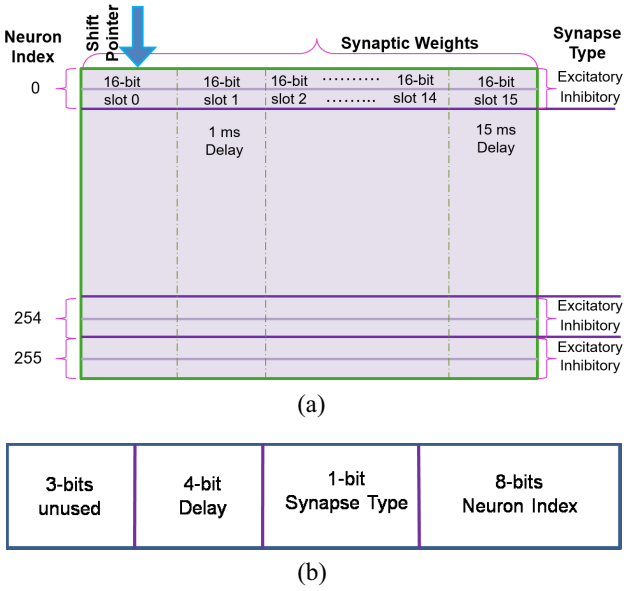

Fig. 10. (a) sPyNNaker ring buffer is at the heart of spike information transfer on the SpiNNaker machine. The first 16 bits of the 32-bit synaptic data word, corresponding to a spike event on the SpiNNaker machine, consists of the synaptic weight data, while (b) the second 16 bits of the 32-bit synaptic data word carry information that guide the placement of the synaptic weight on the ring buffer. Readers may note that for simplicity, we have demonstrated the case for simulation (i.e., solver) time-step of 1 ms.

In our above-mentioned example case ($d_{\text{conn}} = 3$ ms), therefore, if the pointer is pointing currently at the slot 14, then our synaptic weight data will be placed in slot 1, i.e., circular shifted third position from slot 14. Furthermore, if we assume there are "n" excitatory synapses arriving to our single afferent neuron-Y, and all synapses are to be activated after 3 ms, then the resultant synaptic weight that is placed in the ring buffer at the 3-ms delay slot is a linear summation of all the afferent weights (6). Note that for simplicity and demonstration purposes, we have assumed a solver time-step of 1 ms. [The solver time-step is 0.1 ms (aforementioned) in the BG model, and there is an additional "delay extension" mechanism implemented to handle delays over 16-ms time-steps, which will be discussed in a future publication.] The pointer advances one slot to the right in each time-step and the synaptic weight data in the pointed slot is passed on to the neuronal equation solver. Thus, in our example case, 0 is passed for the next two simulation time-steps, i.e., there is no synaptic effect on the neuronal behavior. In the third time-step, the pointer now points to slot 1, the synaptic weight is passed on to the neuronal computation, and the ring buffer location is reset to 0.

Overall, the ring buffer forms the backbone for synapse implementation on SpiNNaker. Moreover, it provides an effective algorithm to incorporate synaptic delays in real time, thus alleviating the need for complex mathematical formulations to achieve the same.

B. Bootstrap Analyses of Spike Counts

To test whether two models produced the same number of spikes, we applied a *studentized, bootstrapped test of the equality of means* of two distributions of spike counts obtained by running our simulations $n = 30$ times. The test follows [19, Algorithm 16.2] and has the null hypothesis, H_0: *the means are the same*. Consider two samples, $\mathbf{z}$ and $\mathbf{y}$ (both of size n); which are sample spike counts generated by (for example) the STN in the SpiNNaker and SpineML models. The *observed value* of this test, $t(\mathbf{x})$, is a studentized (meaning the variances are accounted for) difference of the means of $\mathbf{z}$ and $\mathbf{y}$, given by

$$t(\mathbf{x}) = \frac{\bar{z} - \bar{y}}{\sqrt{\sigma_z^2/n + \sigma_y^2/n}}$$

where $\mathbf{x}$ is the combined sample formed by joining $\mathbf{z}$ and $\mathbf{y}$, σ_z and σ_y are the standard deviations of $\mathbf{z}$ and $\mathbf{y}$, and $\bar{z}$ and $\bar{y}$ are the arithmetic means of $\mathbf{z}$ and $\mathbf{y}$.

A set of resamples is now made from $\mathbf{z}$ and $\mathbf{y}$ after applying a transformation that assumes the null hypothesis is true. The transformations are defined as

$$\tilde{\mathbf{z}} = \mathbf{z} - \bar{z} + \bar{x}; \quad \tilde{\mathbf{y}} = \mathbf{y} - \bar{y} + \bar{x}$$

where $\bar{x}$ is the mean of $\mathbf{x}$. This shifts $\mathbf{z}$ and $\mathbf{y}$ to force their means to be equal. $\mathbf{z}^*$ and $\mathbf{y}^*$ are individual resamples from $\tilde{\mathbf{z}}$ and $\tilde{\mathbf{y}}$. The studentized difference of the means of the resamples is computed

$$t(\mathbf{x}^*) = \frac{\bar{z}^* - \bar{y}^*}{\sqrt{\sigma_z^{*2}/n + \sigma_y^{*2}/n}}.$$

If the original means of $\mathbf{z}$ and $\mathbf{y}$ were genuinely very close, then $\tilde{\mathbf{z}}$ and $\tilde{\mathbf{y}}$ will not have been shifted very much and it is likely that $t(\mathbf{x}^*)$ will exceed $t(\mathbf{x})$ with probability around 0.5. If they were not close, and the mean($\mathbf{z}$) $\gg$ mean($\mathbf{y}$), then very few $t(\mathbf{x}^*)$ will exceed $t(\mathbf{x})$. We made 10 000 $\mathbf{x}^*$ resamples; the proportion of those for which $t(\mathbf{x}^*) \geq t(\mathbf{x})$ is the ASL. The smaller ASL is, the less probable is H_0, and the more significant is the difference of the means.

The test makes no assumption about the shape of the distributions which generated the samples, but it does assume that $\bar{z} > \bar{y}$.

Author Contribution

B. Sen-Bhattacharya designed the model layout, implemented the model on SpiNNaker, and generated SpiNNaker-based model simulation results. S. James implemented the model on SpineML, generated all SpineML-based model simulation results, and carried out statistical test for model comparison. O. Rhodes made the performance analysis for time requirements. I. Sugiarto built the power measurement equipment and performed power analysis of the model simulation on SpiNNaker. These four authors also generated the figures. A. Rowley and A. B. Stokes supported with sPyNNaker, and implementation and validation of the SpiNNaker-based model. K. Gurney advised on BG model design and implementation. S. B. Furber provided advice, guidance, and support for model implementation on SpiNNaker. All the authors contributed to write this paper.

ACKNOWLEDGMENT

SpiNNaker has been 15 years in conception and ten years in construction, and many folks in Manchester, U.K., and in our various collaborating groups around the world have contributed to get the project to its current state. The authors would like to thank all of these contributions.

REFERENCES

[1] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker project," *Proc. IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.

[2] J. W. Mink, "The basal ganglia: Focused selection and inhibition of competing motor programs," *Progr. Neurobiol.*, vol. 50, no. 4, pp. 381–425, 1996.

[3] K. Gurney, T. J. Prescott, and P. Redgrave, "A computational model of action selection in the basal ganglia. I. A new functional anatomy," *Biol. Cybern.*, vol. 84, no. 6, pp. 401–410, 2001.

[4] S. B. Furber *et al.*, "Overview of the SpiNNaker system architecture," *IEEE Trans. Comput.*, vol. 62, no. 12, pp. 2454–2467, Dec. 2013.

[5] E. Painkras *et al.*, "SpiNNaker: A 1-W 18-core system-on-chip for massively-parallel neural network simulation," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1943–1953, Aug. 2013, doi: 10.1109/JSSC.2013.2259038.

[6] K. Gurney, T. J. Prescott, and P. Redgrave, "A computational model of action selection in the basal ganglia. II. Analysis and simulation of behaviour," *Biol. Cybern.*, vol. 84, no. 6, pp. 4011–4423, 2001.

[7] M. D. Humphries and K. N. Gurney, "A pulsed neural network model of bursting in the basal ganglia," *Neural Netw.*, vol. 14, nos. 6–7, pp. 845–863, 2001.

[8] M. D. Humphries and K. N. Gurney, "The role of intra-thalamic and thalamocortical circuits in action selection," *Networks*, vol. 13, no. 1, pp. 131–156, 2002.

[9] M. D. Humphries, R. D. Stewart, and K. N. Gurney, "A physiologically plausible model of action selection and oscillatory activity in the basal ganglia," *J. Neurosci.*, vol. 26, no. 50, pp. 12921–12942, 2006.

[10] K. Gurney, T. J. Prescott, J. R. Wickens, and P. Redgrave, "Computational models of the basal ganglia: From robots to membranes," *Trends Neurosci.*, vol. 27, no. 8, pp. 453–459, 2004.

[11] T. J. Prescott, F. M. M. Gonzalez, K. Gurney, M. D. Humphries, and P. Redgrave, "A robot model of the basal ganglia: behavior and intrinsic processing," *Neural Netw.*, vol. 19, no. 1, pp. 31–61, 2006.

[12] M. D. Humphries, R. Wood, and K. Gurney, "Dopamine-modulated dynamic cell assemblies generated by the GABAergic striatal microcircuit," *Neural Netw.*, vol. 22, no. 8, pp. 1174–1188, 2009.

[13] A. B. Stokes *et al.* (2016). *sPyNNaker 3.0.0*. [Online]. Available: https://github.com/SpiNNakerManchester/sPyNNaker/releases/tag/3.0.0

[14] C. Liu *et al.*, "Dynamical analysis of parkinsonian state emulated by hybrid Izhikevich neuron models," *Commun. Nonlin. Sci. Numer. Simulat.*, vol. 28, nos. 1–3, pp. 10–26, 2015.

[15] C. M. Thibeault and N. Srinivasan, "Using a hybrid neuron in physiologically inspired models of the basal ganglia," *Front. Comput. Neurosci.*, vol. 7, no. 88, pp. 1–17, 2013.

[16] P. Richmond, A. Cope, K. Gurney, and D. J. Allerton, "From model specification to simulation of biologically constrained networks of spiking neurons," *Neuroinformatics*, vol. 12, no. 2, pp. 307–323, Apr. 2014.

[17] A. J. Cope, P. Richmond, S. S. James, K. Gurney, and D. J. Allerton, "SpineCreator: A graphical user interface for the creation of layered neural models," *Neuroinformatics*, vol. 15, no. 1, pp. 25–40, 2017.

[18] A. J. Cope, P. Richmond, and S. S. James. (2015). *SpineCreator*. [Online]. Available: RRID SCR_015641

[19] B. Efron and R. J. Tibshirani, *An Introduction to Bootstrap*, Chapman and Hall, 1994.

[20] I. Sugiarto *et al.*, "Profiling a many-core neuromorphic platform," in *Proc. 11th IEEE Int. Conf. Appl. Inf. Commun. Technol. (AICT)*, vol. 2, 2017, pp. 42–47.

[21] D. E. Oorschot, "Total number of neurons in the neostriatal, pallidal, subthalamic, and substantia nigral nuclei of the rat basal ganglia: A stereological study using the cavalieri and optical disector methods," *J. Comp. Neurol.*, vol. 366, no. 4, pp. 580–599, 1996.

[22] T. Koós and J. M. Tepper, "Inhibitory control of neostriatal projection neurons by GABAergic interneurons," *Nature*, vol. 2, no. 5, pp. 467–472, 1999.

[23] N. Mallet, B. Ballion, C. L. Moine, and F. Gonon, "Cortical inputs and GABA Interneurons imbalance projection neurons in the striatum of parkinsonian rats," *J. Neurosci.*, vol. 26, no. 14, pp. 3875–3884, 2006.

[24] J. T. Moyer, J. A. Wolf, and L. H. Finkel, "Effects of dopaminergic modulation on the integrative properties of the ventral striatal medium spiny neuron," *J. Neurophysiol.*, vol. 98, no. 6, pp. 3731–3748, 2007.

[25] G. Radnikow and U. Misgeld, "Dopamine D_1 receptors facilitate $GABA_A$ synaptic currents in the rat Substantia Nigra pars reticulata," *J. Neurosci.*, vol. 18, no. 6, pp. 2009–2016, 1998.

[26] D. L. Cameron and J. T. Williams, "Dopamine D1 receptors facilitate transmitter release," *Nature*, vol. 366, no. 6453, pp. 344–347, 1993.

[27] Y. Kawaguchi, C. J. Wilson, S. J. Augood, and P. C. Emson, "Striatal interneurones: Chemical, physiological and morphological characterization," *Trends Neurosci.*, vol. 18, no. 12, pp. 527–535, 1995.

[28] S. Taverna, E. Llijic, and D. J. Surmeier, "Recurrent collateral connections of striatal medium spiny neurons are disrupted in models of Parkinson's disease," *J. Neurosci.*, vol. 28, no. 21, pp. 5504–5512, 2008.

[29] S. Taverna, B. Canciani, and C. M. Pennartz, "Membrane properties and synaptic connectivity of fast-spiking interneurons in rat ventral striatum," *Brain Res.*, vol. 1152, pp. 49–56, Jun. 2007.

[30] D. Jaegar and H. Kita, "Functional connectivity and integrative properties of globus pallidus neurons," *Neuroscience*, vol. 198, pp. 44–53, Dec. 2011.

[31] J. M. Tepper and J. P. Bolam, "Functional diversity and specificity of neostriatal Interneurons," *Current Opin. Neurobiol.*, vol. 14, pp. 685–692, Dec. 2004.

[32] P. C. Emson, H. J. Waldvogel, and R. L. Faull, "Neurotransmitter receptors in the basal ganglia," in *Handbook of Basal Ganglia Structure and Function*. Amsterdam, The Netherlands: Academic Press, 2010.

[33] P. Mailly, S. Charpier, A. Menetrey, and J.-M. Deniau, "Three-dimensional organization of the recurrent axon collateral network of the substantia nigra pars reticulata neurons in the rat," *J. Neurosci.*, vol. 23, no. 12, pp. 5247–5257, 2003.

[34] E. M. Izhikevich, "Hybrid spiking models," *Philosoph. Trans. Roy. Soc. A*, vol. 368, no. 1930, pp. 5061–5070, 2010.

[35] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Trans. Neural Netw.*, vol. 14, no. 6, pp. 1569–1572, Nov. 2003.

[36] W. Paulus and J. C. Rothwell, "Membrane resistance and shunting inhibition: Where biophysics meets state-dependent human neurophysiology," *J. Physiol.*, vol. 594, no. 10, pp. 2719–2728, 2016.

[37] P. Jonas and G. Buzsaki, "Neural inhibition," *Scholarpedia*, vol. 2, no. 9, p. 3286, 2007.

[38] K. Kaila, "Ionic basis of $GABA_A$ receptor channel function in the nervous system," *Progr. Neurobiol.*, vol. 42, no. 4, pp. 489–537, 1994.

[39] J. P. Bolam, J. J. Hanley, P. A. C. Booth, and M. D. Bevan, "Synaptic organisation of the basal ganglia," *J. Anatomy*, vol. 196, pp. 527–542, May 2000.

[40] D. Centonze *et al.*, "Receptor subtypes involved in the presynaptic and postsynaptic actions of dopamine on striatal interneurons," *J. Neurosci.*, vol. 23, no. 15, pp. 6245–6254, 2003.

[41] S. J. Cragg, J. Boufreton, Y. Xue, J. P. Bolam, and M. D. Bevan, "Synaptic release of dopamine in the subthalamic nucleus," *Eur. J. Neurosci.*, vol. 20, no. 7, pp. 1788–1802, 2004.

[42] J. T. Trevitt, J. Morrow, and J. F. Marshall, "Dopamine manipulation alters immediate-early gene response of striatal parvalbumin interneurons to cortical stimulation," *Brain Res.*, vol. 1035, no. 1, pp. 41–50, 2005.

[43] M. D. Bevan, N. E. Hallworth, and J. Baufreton, "GABAergic control of the subthalamic nucleus," in *Progress in Brain Research*, vol. 160, Elsevier, 2007.

[44] K. Watanabe, T. Kita, and H. Kita, "Presynaptic actions of D2-like receptors in the rat cortico-striato-globus pallidus disynaptic connection in vitro," *J. Neurophysiol.*, vol. 101, no. 2, pp. 665–671, 2009.

[45] O. Mamad, C. Delaville, W. Benjelloun, and A. Benazzouz, "Dopaminergic control of the globus pallidus through activation of D2 receptors and its impact on the electrical activity of subthalamic nucleus and substantia nigra reticulata neurons," *PLoS ONE*, vol. 10, no. 3, pp. 1–16, 2015.

[46] A. Tomkins, E. Vasilaki, C. Beste, K. Gurney, and M. D. Humphries, "Transient and steady-state selection in the striatal microcircuit," *Front. Comput. Neurosci.*, vol. 7, pp. 1–16, Jan. 2014.

[47] S. Furber, "Large-scale neuromorphic computing systems," *J. Neural Eng.*, vol. 13, no. 5, pp. 1–14, 2016, doi: 10.1088/1741-2560/13/5/051001.

[48] A. P. Davison *et al.*, "PyNN: A common interface for neuronal network simulators," *Front. Neuroinformat.*, vol. 2, no. 11, p. 204, 2009, doi: 10.3389/neuro.11.011.2008.

[49] A. J. Cope and S. S. James. (2015). *SpineML_2_Brahms*. [Online]. Available: http://RRIDSCR_015640
[50] G. E. Alexander, M. R. DeLong, and P. L. Strick, "Parallel organization of functionally segregated circuits linking basal ganglia and cortex," *Annu. Rev. Neurosci.*, vol. 9, pp. 357–381, Mar. 1986.
[51] H. F. Kim and O. Hikosaka, "Parallel basal ganglia circuits for voluntary and automatic behaviour to reach reward," *Brain*, vol. 138, pp. 1776–1800, Jul. 2015.
[52] T. Sharp and S. Furber, "Correctness and performance of the SpiNNaker architecture," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2013, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6706988/
[53] J. Conradt, F. Galluppi, and T. C. Stewart, "Trainable sensorimotor mapping in a neuromorphic robot," *Robot. Autom. Syst.*, vol. 71, pp. 60–68, Sep. 2015.
[54] I. Sugiarto and J. Conradt, "Factor graph inference engine on the SpiNNaker neural computing system," in *Proc. Int. Conf. Artif. Neural Netw. (ICANN)*, 2014, pp. 161–168.

Basabdatta Sen-Bhattacharya, photograph and biography not available at the time of publication.

Sebastian James, photograph and biography not available at the time of publication.

Oliver Rhodes, photograph and biography not available at the time of publication.

Indar Sugiarto, photograph and biography not available at the time of publication.

Andrew Rowley, photograph and biography not available at the time of publication.

Alan B. Stokes, photograph and biography not available at the time of publication.

Kevin Gurney, photograph and biography not available at the time of publication.

Steve B. Furber, photograph and biography not available at the time of publication.