Proceedings of the
8th Graz Brain-Computer Interface Conference 2019 DOI: 10.3217/978-3-85125-682-6-31

A SIMULATED ENVIRONMENT FOR STUDYING PARTIAL OBSERVABILITY IN NOVEL ADAPTIVE DEEP BRAIN STIMULATION

Sebastián Castaño-Candamil[1], Mara Vaihinger[1], Michael Tangermann[1,2]

[1]Brain State Decoding Lab, Department of Computer Science,
BrainLinks-BrainTools, University of Freiburg, Germany
[2]Autonomous Intelligent Systems, Department of Computer Science,
University of Freiburg, Germany

E-mail: sebastian.castano|michael.tangermann@blbt.uni-freiburg.de

ABSTRACT: Adaptive deep brain stimulation (aDBS) can profit from data-driven approaches developed for BCIs. These aDBS systems improve upon the constant DBS in terms of efficiency and side effects amelioration by taking the ongoing brain state into consideration.

The environment controlled by aDBS is governed by partial observability, rendering classic control strategies suboptimal. In this regard, development of novel approaches is critical for improved aDBS therapy. However, early stage aDBS development is a difficult endeavor, given the lack of suitable development platforms.

In our contribution, we present a simulated environment that allows to modularly embed different surrogates of key challenges found in the aDBS problem. Specifically, we will focus on partial observability stemming from non-stationary dynamics and noisy state representations. Our simulations are used to analyze representative reinforcement learning approaches regarding their ability to cope with the partial observability.

To allow reproducibility and encourage adoption of our approach, the source code of our experiments is made available online.

INTRODUCTION

Deep brain stimulation (DBS) has been established as standard clinical treatment for movement disorders, such as Parkinson's disease (PD) and essential tremor (ET) [1, 2]. In addition, it is investigated to provide symptom relief in several neuropsychiatric diseases such as obsessive compulsive disorder (OCD) and major depressive disorder (MDP) [3, 4].

The stimulation characteristics and thus the efficiency of DBS treatment can be shaped by a number of parameters, e.g., electrode contacts used to deliver the electric stimulation pulses, the shape, width and amplitude of pulses and the frequency of these pulses. In a standard clinical setting, DBS parameters are determined by a highly trained clinician and are kept constant until the next consultation. This manual adaptation, performed a few times per year, will ideally account for initial post-surgical transient effects and long-term variations caused by disease progress and DBS-induced plasticity changes. However, such a constant DBS (cDBS) strategy can not cope with changes occurring on much shorter timescales. As a result, patients undergoing cDBS therapy are prone to acute and chronic motor- and neuropsychiatric side-effects, such as speech disorders, dysarthria, depression, emotional disinhibition, and paresthesias [5–7].

Closed loop strategies for DBS: Fortunately, closed-loop adaptive DBS (aDBS) provides a promising approach for tackling the shortcomings of cDBS strategies [8, 9]. Closed-loop aDBS systems provide stimulation as a function of symptoms and DBS-induced side-effects surrogates, extracted directly from brain signals, and termed neural markers (NMs) [10, 11]. Such NMs, however, are highly contaminated by background activity and are co-modulated by multiple brain processes; thus, providing only a partial representation of the real neural state of the patient [9]. Furthermore, despite improving upon cDBS, aDBS systems usually implement control strategies—such as threshold-based and proportional control—that neglect time dynamics.

Alternatively, more complex strategies have attempted to use latent neural dynamics as a potential source of information for improving aDBS efficacy [12–15]. While such dynamics can originate from inherent temporal brain activity, others can be explicitly associated with external factors such as medication intake, activity of daily living (ADL), and circadian rhythm. Another major source of non-stationary dynamics is the so-called DBS washout effect, describing the persistent clinical effect of DBS after stimulation withdrawal [16].

Data-driven approaches for dynamics-aware aDBS: Many of the diseases treated with DBS, as PD and MDP, are characterized by a remarkably heterogeneous phenotype [17, 18], where group studies are unable to deliver NMs and stimulation strategies that are universally suitable. In contrast, data-driven optimization of dynamics-aware control strategies offers a promising approach for obtaining effective and efficient aDBS systems. In this regard, Kumar and colleagues [19] presented a proof-of-concept *in-vitro* study using a tabular reinforcement learning (RL) strategy to control a neural network. Such classic RL strategies assume that the underlying controlled system (*environment*, in RL literature) is *Markovian* and *fully observable*, i.e., predictions of future states of the environment depend solely on the current observation and this observation should offer a full representation of the environment's state. Given that in aDBS both assumptions are not fulfilled, classic RL strategies might deliver a sub-optimal control strategy (*policy*).

Strictly speaking, the non-stationary dynamics seen in aDBS are a consequence of partial observability: if information about non-stationary sources is included in the state representation (e.g., medication intake or ADL context), then the environment could be considered stationary. However, such information is usually not available. Consequently, we use the term *partial observability* to cover non-stationary dynamics and noisy NMs, hereafter.

Development platforms for aDBS: Working directly with patients is an expensive and strongly constrained endeavor (safety regulations), and *in-vitro* development protocols are relatively expensive and may suffer from oversimplifying assumptions regarding the structure of the underlying neural network. For these reasons, *in-silico* frameworks have been widely utilized in early aDBS development stages [20–22].

With our current contribution we introduce a novel *in-silico* approach. We adopt a modified version of an environment used in the standard RL testbench *openAI Gym*[23] and show, how partial observability properties of aDBS can be explicitly embedded into a RL task, thus making partial observability a benchmarkable challenge. Finally, we provide a comparison of state-of-the-art RL algorithms that deliver a more efficient control strategy than classic aDBS approaches under this partially observable environment.

METHODS

The core concept of RL is to learn how to control an environment alone from interactions with it. During the learning phase, the decision making *agent* continuously improves its control *policy* based on the *reward* it gains by interacting with the—potentially unknown—*environment*. In the aDBS context, the mapping between the RL components can be defined as: agent↔DBS controller, environment↔neural system, policy↔stimulation strategy, action↔apply a (parameterized) stimulation, and reward↔symptom suppression / side effects. This mapping can be formalized by defining an aDBS system as a Markov decision process, as follows.

Closed-loop aDBS as a Markov decision process: A closed-loop aDBS system can be defined as a Markov decision process $\mathscr{M} = \langle \mathscr{S}, \mathscr{A}, T, R \rangle$, where the space $\mathscr{S}$ is formed by all possible motor states, $\mathscr{A}$ is the set of possible stimulation parameters, $T : (s_k, s_{k+1}, a) \mapsto p(s_{k+1}|s_k, a_k) \in [0,1]$ is the probability distribution over brain state transitions, such that applying the stimulation parameter $a_k \in \mathscr{A}$ in the brain state $s_k \in \mathscr{S}$ at time point k leads to a new brain state $s_{k+1} \in \mathscr{S}$ at time $k+1$, and $R : (s_k, s_{k+1}) \mapsto R(s_k, s_{k+1}) \in [a,b]$ is the reward function (bounded by $\{a,b\} \in \mathbb{R}$) obtained by transitioning from s_k to s_{k+1}. In aDBS, a reward may express, e.g., the amelioration of symptoms or the suppression of DBS-induced side effects.

Partial observability in aDBS:

Noise regimes in NMs: Brain imaging techniques, such as electrocorticographic recordings or local field potential recordings from deep brain electrodes, are contaminated by measurement noise and background activity and are co-modulated by several, possibly independent, neural processes. In PD, the beta-band power of local field potentials recorded from the subthalamic nucleus (STN) is a widely used NM. However, it is not only determined by the symptom state, but is also modulated by motor preparation and execution[24, 25]—similar to cortical beta band power—. These characteristics render many extracted NMs highly noisy and thus they contribute to partial observability found in aDBS.

Washout-induced non-stationary dynamics: The washout effect observed in DBS contributes to partial observability by generating non-stationary dynamics on multiple timescales. For example in PD, DBS washout effects w.r.t. axial symptoms that influence gait or speech, can span from minutes to several hours, whereas washout w.r.t. rigor, tremor, and bradykinesia, typically lasts seconds only. In the Markov decision process defined above, a washout phase amounts to state transition distribution T that not only depends on the current stimulation, but also on the history thereof.

As some classes of control algorithms have not been designed to cope well with partial observability, it is important to investigate its effects upon the effectiveness and efficiency of aDBS control strategies. However, in *in-vivo* or *in-vitro* scenarios, it is difficult to analyze the specific impact of such dynamics individually. A surrogate environment provides the possibility to model those aDBS-specific challenges explicitly.

The flappingBird environment: We adapted the *FlappyBird* environment provided in the openAI gym platform to incorporate challenges of aDBS. We term the adapted environment the *continuous FlappyBird* (CFB). An agent's goal in this environment is to fly through horizontal tunnels that constantly pass by, as a gravity force pulls the agent downwards.Two main criteria were considered for selecting CFB as our surrogate environment: First, dimensionality of the state and action space is similar to the aDBS problem [26], and second, the model's engine provides a computationally inexpensive way of modifying the environment dynamics.

State representation: The state of the CFB environment is given by a 7-dimensional signal. It comprises 1) agent vertical position, 2) agent vertical velocity 3) and 4) bottom and top vertical position of the current tunnel, 5) agent's distance to the next tunnel and 6) and 7) bottom and top position of the next tunnel. The dimensionality of the CFB state space is similar to that of simple aDBS setups: $\mathscr{S}$ is usually represented by a small number of power features extracted from local field potential signals [27–29]. Assuming that each of the bilaterally implanted DBS electrodes has 4 contacts, $\mathscr{S}$ results in an 8-dimensional representation.

Actions: The binary action space comprises two actions: vertical thrust and no vertical thrust. In a simplified approach to aDBS, stimulus amplitude parameter can also be defined as binary, i.e., DBS-on/off. Note that in more advanced setups, the action space might be continuous (if a continuous amplitude control is desired) or multidimensional (if other DBS parameters like stimulation frequency or stimulating contacts are considered).

Reward signal: Designing a good reward signal is difficult and problem specific. In aDBS, it requires a tradeoff between at least the amelioration of PD-related symptoms and stimulation-induced side effects. In CFB, we define the reward signal as a function of the proximity with the center of a tunnel, for each time point k:

$$R(s_k, s_{k+1}) = \begin{cases} 0.1 & \text{if in } s_{k+1} \text{ agent is inside tunnel} \\ -0.4 & \text{if in } s_{k+1} \text{ agent outside tunnel} \\ -0.9 & \text{if in } s_{k+1} \text{ top or bottom is hit} \end{cases}$$

Partial observability in the CFB environment:

Noise in the state representation: Noisy state measurements are embedded in the environment by adding zero-mean Gaussian noise to each feature describing the state. The standard deviation of each noise source is defined individually per feature as $\sigma_f = \xi \cdot (l_f^u - l_f^d)$, where ξ denotes the noise level and the interval $[l_f^d, l_f^u]$ define domain of feature f. This modified version of CFB is called **CFB-Nξ** in the following.

History dependent action effect: The washout effect is simulated as a sustained aftereffect of each thrust action, and termed CFB-H. It is implemented using an action history which considers (at maximum) the last 100 thrusts. Specifically, the decaying thrust T_k^d at a time point k after a sequence of N_{thrust} thrust actions in the last 100 time steps is defined as: $T_k^d = T_{k-1}^d - \frac{T}{N_{thrust}/3}$, where T represents the thrust generated by a single *thrust* action. The constants selected here for the decaying thrust function are based on studies reporting ratios between accumulated stimulation N_{thrust} and washout duration in a range of 8:1 to 2:1 [30, 31]. The horizon is limited to a history of 100 time steps to ensure at least a limited level of controllability in the CFB-H environment.

EXPERIMENTAL SETUP

Choice of RL algorithms: Three model-free RL strategies were chosen as base algorithms, from the three main method families in RL: Value-function based, policy gradient, and actor critic. The selection of specific methods involved the following criteria: First, we considered the reported performance across multiple RL tasks in the OpenAI benchmark[1]. Second, we took into account the scientific impact of each algorithm within the RL community, as measured by the number of citations of the corresponding papers, their publication date, and number of appearances in review studies. The resulting collection of representative RL base algorithms, each using a feed-forward (FF) neural network, comprises: 1) Deep Q-Learning (DQN) with experience replay and a target network [32], an off-policy value function based method, 2) Advantage Actor-Critic (A2C), a 1-step advantage actor-critic method [33], and 3) Proximal Policy Optimization (PPO), a 1-step advantage actor-critic method focusing on an improved policy gradient estimate [34].

For comparison, we have also included a simple reactive agent, designed to resemble a threshold-based control strategy. It applies thrust whenever the agent finds itself below the center of a tunnel and stops when it is above the tunnel.

RL approaches to non-stationary MDP: By their design, the RL base algorithms DQN+FF, A2C+FF and PPO+FF can not be expected to deal well with partial observability of non-stationary origin. However, they can be equipped with the ability to consider a (potentially infinite) state (or state-action) history as proposed by [35–38]. For this reasons we have extended the DQN, A2C, and PPO models by recurrent networks implemented using either gated recurrent units (GRUs) or long short term memory (LSTM) units [39].

As a result, we could benchmark the following nine RL approaches: DQN+FF, DQN+LSTM, DQN+GRU, A2C+FF, A2C+LSTM, A2C+GRU, PPO+FF, PPO+LSTM, and PPO+GRU.

Benchmark design:

Model architectures and hyperparameter optimization: For all models, hyperparameters were optimized using the sequential model-based configuration framework introduced in [40]. The source code including all parameter details used to optimize and train our agents and the resulting architectures is provided online[2].

Training stage: A population of twenty RL agents per method was trained, each trained in an individual instance of the same RL task, but initialized with different random seeds. All agents were trained over two million interactions in the environments CFB-N and CFB-H. An Adam optimizer [41] updated a model's parameters every 64 interactions. The performance of a learning agent during training is evaluated in an independent test environment every 300 parameter updates. The performance reported corresponds to the average reward obtained by each population of agents.

Testing stage: After training, each agent population is tested in twenty randomly initialized environments during 50k interactions. The reported performance corresponds to the average reward for each agent population.

RESULTS

Performance in CFB-H environments:

[1] https://github.com/openai/baselines-results

[2] https://github.com/mVaihinger/RLAgentsFlappyBird

The training performance of the different agents in the CFB-H environments are depicted in Figure 1. Only the A2C agents consistently achieved a better performance than the reactive agents. PPO-FF achieved also a similar performance, however, it was less stable throughout training. All DQN-based agents had a much slower convergence rate and stayed considerably below the performance of the simple reactive agents.

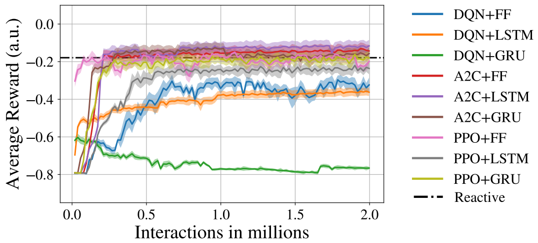

Figure 1: Time courses of training performance over two million steps in the CFB-H task.

Consistent with the training stage, Figure 2 shows that DQN agents yielded the worst performance in the test stage, whereas agents based on A2C and PPO better than the reactive agents on average. Overall, agents based on LSTM units achieved the best performance.

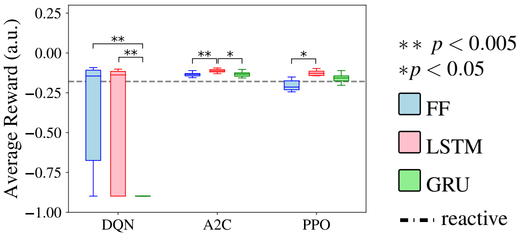

Figure 2: Boxplot of RL agent's performance in the unseen CFB-H test environments. Statistical significance was tested with the Wilcoxon ranksum using Bonferroni correction.

Performance in CFB-N environments: Figure 3 shows the training performance of all agents in CFB-N environments. For the RL-methods, the main difference elicited by varying noise levels is the final training performance, while the convergence rate was rather unaffected by noise. Among all, A2C and PPO agents showed the greatest sample efficiency, as their performance improved the fastest in early stages of training, with A2C showing the most stable performance throughout training. While A2C-LSTM showed the best performance, even under strong noise regimes ($\xi < 0.5$), it is interesting to see, that A2C+FF could still cope quite well with the noise, while GRU-based models yielded the worst performance. These observations also hold during test stage, shown in Figure 4.

DISCUSSION

We have introduced a simulated environment to support early development stages of data-driven aDBS strategies based on RL. Specifically, the environment is designed to study approaches coping with partial observability properties. In addition, we have benchmarked representative RL methods, that have been modified to cope with the challenges faced by an aDBS system.

Suitability of considered methods for partially observable environments: The comparative analysis presented was motivated by the hypothesis that models aimed at capturing long term dependencies yield a higher end performance in partial observable environments, compared to classic approaches. We have shown that using LSTM- and GRU-based models does not consistently improve end performance compared to agents based on classic FF networks; however, LSTM-based models enable higher sample efficiency, as proved by the faster convergence of such models in early stages of training. We have also observed that the major performance difference is caused by the RL method chosen, and not by the type of recurrent units used in them.

CFB as a development environment of RL-aDBS: Although it is not possible to establish a one to one correspondence between a real aDBS environment and the CFB environment used, our framework allows to study specific characteristics of aDBS in early stages of control algorithm development, when physiological and functional constraints and clinical interpretability are not critical. A key feature of our contribution is the flexibility to explicitly embed key challenges found in aDBS in a modular fashion. In the present contribution, we have studied precise non-stationary dynamics caused by DBS washout, as well as noisy state representations. However, our framework can easily include other major sources of partial observability such as circadian rhythm variations, medication induced changes, among others. The only prerequisite is an appropriate scaling of time constants, as we have exemplified with the CFN-H environment.

In conclusion, our framework provides a cost efficient platform for early stage development of novel aDBS strategies before accessing more complicated setups, as physiologically-motivated simulations and, as a final goal, patients.

ACKNOWLEDGMENTS

This work was supported by BrainLinks-BrainTools Cluster of Excellence funded by the German Research Foundation (DFG, EXC1086), by the Federal Ministry of Education and Research (BMBF, 16SV8012), and by the state of Baden-Württemberg, and the DFG through grants bwHPC and INST 39/963-1 FUGG.

REFERENCES

[1] Baizabal-Carvallo J, Kagnoff M, Jimenez-Shahad J, Fekete R, Jankovic J. The safety and efficacy of thalamic deep brain stimulation in essential tremor: 10 years and beyond. J Neurol Neurosurg Psychiatry. 2014;85(5):567–72.

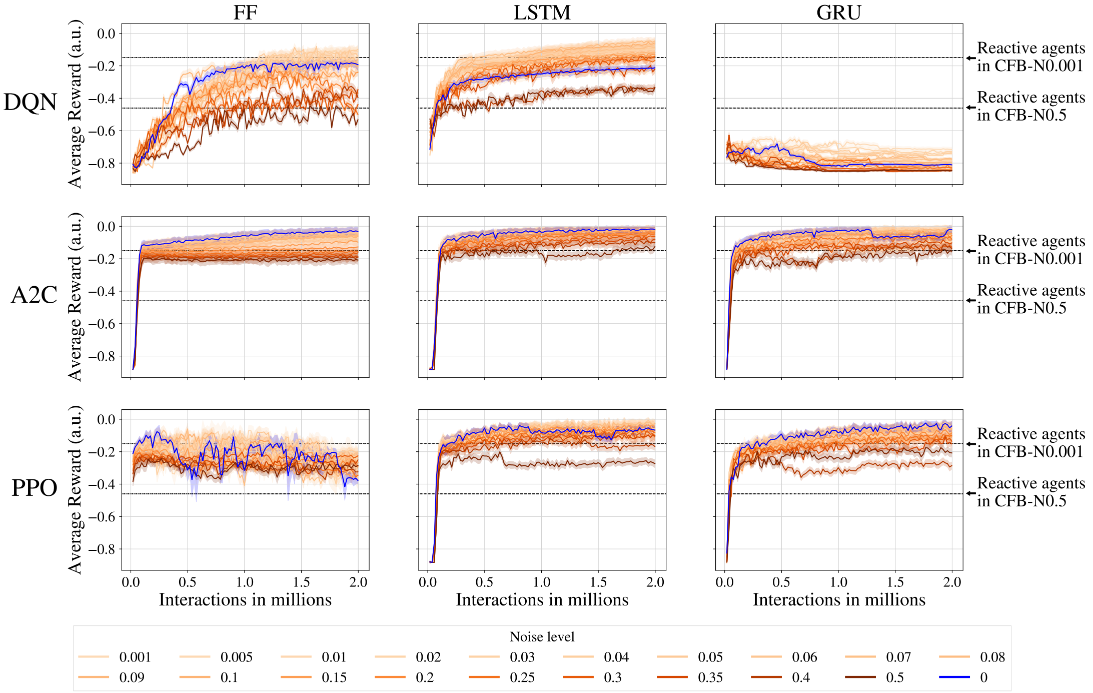

Figure 3: Time courses of the training performance over two million steps in the CFB-N task for several noise levels.

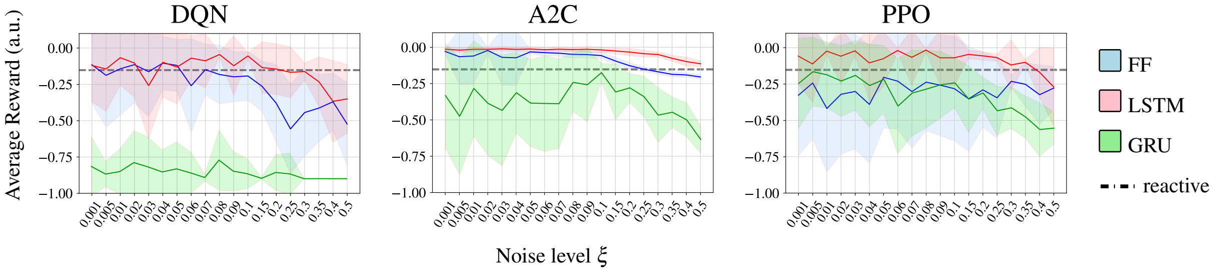

Figure 4: RL agent's performance in the unseen CFB-N test scenarios for several noise levels.

[2] Rodriguez-Oroz MC, Obeso JA, Lang AE, *et al.* Bilateral deep brain stimulation in Parkinson's disease: A multicentre study with 4 years follow-up. Brain. 2005;128(10):2240–49.

[3] Widge AS, Malone Jr DA, Dougherty DD. Closing the loop on deep brain stimulation for treatment-resistant depression. Frontiers in neuroscience. 2018;12:175.

[4] Coenen VA, Schlaepfer TE, Goll P, *et al.* The medial forebrain bundle as a target for deep brain stimulation for obsessive-compulsive disorder. CNS spectrums. 2017;22(3):282–289.

[5] Witt K, Daniels C, Volkmann J. Factors associated with neuropsychiatric side effects after STN-DBS in Parkinson's disease. Parkinsonism & related disorders. 2012;18:S168–S170.

[6] Little S, Tripoliti E, Beudel M, *et al.* Adaptive deep brain stimulation for Parkinson's disease demonstrates reduced speech side effects compared to conventional stimulation in the acute setting. Journal of Neurology, Neurosurgery & Psychiatry. 2016;87(12):1388–1389.

[7] Castrioto A, Lhommée E, Moro E, Krack P. Mood and behavioural effects of subthalamic stimulation in Parkinson's disease. The Lancet Neurology. 2014;13(3):287 –305.

[8] Kühn AA, Volkmann J. Innovations in deep brain stimulation methodology: Innovations in DBS Methodology. Movement Disorders. 2017;32(1):11–19.

[9] Little S, Brown P. What brain signals are suitable for feedback control of deep brain stimulation in Parkinson's disease?: Brain signals for control of DBS in PD. Annals of the New York Academy of Sciences. 2012;1265(1):9–24.

[10] Beudel M, Brown P. Adaptive deep brain stimulation in parkinson's disease. Parkinsonism & related disorders. 2016;22:S123–S126.

[11] Kuo C-H, White-Dzuro GA, Ko AL. Approaches to closed-loop deep brain stimulation for movement disorders. Neurosurgical focus. 2018;45(2):E2.

[12] Haddock A, Velisar A, Herron J, Bronte-Stewart H, Chizeck HJ. Model predictive control of deep brain stimulation for Parkinsonian tremor. In: Neural Engineering (NER), 2017 8th International IEEE/EMBS Conference on. 2017, 358–362.

[13] Adamchic I, Hauptmann C, Barnikol UB, *et al.* Coordinated reset neuromodulation for Parkinson's disease: Proof-of-concept study. Movement disorders. 2014;29(13):1679–1684.

[14] Wang J, Nebeck S, Muralidharan A, Johnson MD, Vitek JL, Baker KB. Coordinated reset deep brain stimulation of subthalamic nucleus produces long-lasting, dose-dependent motor improvements in the 1-methyl-4-phenyl-1, 2, 3, 6-tetrahydropyridine non-human primate model of Parkinsonism. Brain stimulation. 2016;9(4):609–617.

[15] Cagnan H, Pedrosa D, Little S, *et al.* Stimulating at the right time: Phase-specific deep brain stimulation. Brain. 2016;140(1):132–145.

[16] Cooper SE, McIntyre CC, Fernandez HH, Vitek JL. Association of deep brain stimulation washout effects with Parkinson disease duration. JAMA neurology. 2013;70(1):95–99.

[17] Thenganatt MA, Jankovic J. Parkinson disease subtypes. JAMA neurology. 2014;71(4):499–504.

[18] Widge AS, Ellard KK, Paulk AC, *et al.* Treating refractory mental illness with closed-loop brain stimulation: Progress towards a patient-specific transdiagnostic approach. Experimental neurology. 2017;287:461–472.

[19] Kumar SS, Wülfing J, Okujeni S, Boedecker J, Riedmiller M, Egert U. Autonomous optimization of targeted stimulation of neuronal networks. PLoS computational biology. 2016;12(8):e1005054.

[20] Popovych OV, Lysyansky B, Tass PA. Closed-loop deep brain stimulation by pulsatile delayed feedback with increased gap between pulse phases. Scientific reports. 2017;7(1):1033.

[21] Karamintziou SD, Custódio AL, Piallat B, *et al.* Algorithmic design of a noise-resistant and efficient closed-loop deep brain stimulation system: A computational approach. PloS one. 2017;12(2):e0171458.

[22] Karamintziou SD, Deligiannis NG, Piallat B, *et al.* Dominant efficiency of nonregular patterns of subthalamic nucleus deep brain stimulation for Parkinson's disease and obsessive-compulsive disorder in a data-driven computational model. Journal of neural engineering. 2015;13(1):016013.

[23] Brockman G, Cheung V, Pettersson L, *et al.* Openai gym. arXiv preprint arXiv:1606.01540. 2016.

[24] Kühn AA, Williams D, Kupsch A, *et al.* Event-related beta desynchronization in human subthalamic nucleus correlates with motor performance. Brain. 2004;127(4):735–746.

[25] Engel AK, Fries P. Beta-band oscillations—signalling the status quo? Current opinion in neurobiology. 2010;20(2):156–165.

[26] Castaño-Candamil S, Vaihinger M, Tangermann M. A Simulated Environment for Early Development Stages of Reinforcement Learning Algorithms for Closed-Loop Deep Brain Stimulation (under review). In: Engineering in Medicine and Biology Society (EMBC), 2019 41st Annual International Conference of the IEEE. 2019.

[27] Kühn AA, Tsui A, Aziz T, *et al.* Pathological synchronisation in the subthalamic nucleus of patients with Parkinson's disease relates to both bradykinesia and rigidity. Experimental neurology. 2009;215(2):380–387.

[28] Blumenfeld Z, Brontë-Stewart H. High Frequency Deep Brain Stimulation and Neural Rhythms in Parkinson's Disease. Neuropsychology Review. 2015;25(4):384–397.

[29] Neumann W-J, Staub-Bartelt F, Horn A, *et al.* Long term correlation of subthalamic beta band activity with motor impairment in patients with Parkinson's disease. Clinical Neurophysiology. 2017;128(11):2286–2291.

[30] Wingeier B, Tcheng T, Koop MM, Hill BC, Heit G, Bronte-Stewart HM. Intra-operative STN DBS attenuates the prominent beta rhythm in the STN in Parkinson's disease. Experimental neurology. 2006;197(1):244–251.

[31] Bronte-Stewart H, Barberini C, Koop MM, Hill BC, Henderson JM, Wingeier B. The STN beta-band profile in Parkinson's disease is stationary and shows prolonged attenuation after deep brain stimulation. Experimental neurology. 2009;215(1):20–28.

[32] Mnih V, Kavukcuoglu K, Silver D, *et al.* Human-level control through deep reinforcement learning. Nature. 2015;518(7540):529.

[33] Mnih V, Badia AP, Mirza M, *et al.* Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. 2016, 1928–1937.

[34] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347. 2017.

[35] Hausknecht M, Stone P. Deep recurrent Q-learning for partially observable MDPS. CoRR, abs/1507.06527. 2015;7(1).

[36] Schmidhuber J. Reinforcement learning in Markovian and non-Markovian environments. In: Advances in neural information processing systems. 1991, 500–506.

[37] Wierstra D, Foerster A, Peters J, Schmidhuber J. Solving deep memory POMDPs with recurrent policy gradients. In: International Conference on Artificial Neural Networks. 2007, 697–706.

[38] Heess N, Hunt JJ, Lillicrap TP, Silver D. Memory-based control with recurrent neural networks. arXiv preprint arXiv:1512.04455. 2015.

[39] Bakker B. Reinforcement learning with long short-term memory. In: Advances in neural information processing systems. 2002, 1475–1482.

[40] Hutter F, Hoos H, Leyton-Brown K. Sequential model-based optimization for general algorithm configuration. LION. 2011;5:507–523.

[41] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014.