計測自動制御学会東北支部 第 332 回研究集会 (2021.5.26)
資料番号 332-1

スパイキングニューラルネットワークに基づく深層強化学習による脚ロボットの歩行生成と評価

Spiking Neural Network Discovers Energy-efficient Hexapod Motion in Deep Reinforcement Learning

○ 納谷克海, 沓澤京, 大脇大, 林部充宏

○ Katsumi Naya, Kyo Kutsuzawa, Dai Owaki, Mitsuhiro Hayashibe

東北大学

Tohoku University

キーワード： スパイキングニューラルネットワーク (Spiking Neural Network), 深層強化学習 (Deep Reinforcement Learning), エネルギー効率 (Energy efficiency), 6 脚ロボット (Hexapod)

連絡先： 〒 980-8579 仙台市青葉区荒巻字青葉 6–6–01 東北大学 工学研究科 機械・知能系共同棟 503
納谷克海, Tel.: (022)795-6970 Fax.: (022)795-697 E-mail: katsumi.naya.p6@dc.tohoku.ac.jp

1. 緒言

現在，未知の環境から最適な行動を学習する深層強化学習 (Deep Reinforcement Learning: DRL) がゲームや自動運転の分野のみならず，ロボティクスの分野においても注目を浴びている．これに対して, 深層強化学習の有力なアルゴリズムとして Deep Deterministic policy gradient(DDPG) [1], Soft Actor-Critic(SAC) [2] や Twin Delayed Deep Deterministic policy gradient(TD3) [3] が挙げられる．DDPG は計算時間の短さに優位性があり、SAC や TD3 は各アルゴリズムの中で特に高い性能を示すことで知られている．

深層強化学習を，自律移動型ロボットの制御に応用する際には, エネルギー資源が限られていることから, 未知環境への適応制御とエネルギー効率の高い制御の両方が求められる．特に，エネルギー効率の良い行動パターンを得るには工夫が必要となる．その一例として, エージェントの行動に重み係数を掛けた運動のペナルティを報酬に追加するという方法がある．この方法では報酬関数に項を足すだけなのであらゆる DRL アルゴリズムに適用でき，また過学習を防止する効果があると報告されている [4]．しかし，このとき重み係数はある程度大きくないと十分な効果をもたらさないが，大きすぎると全く動かないという局所解に陥ってしまう．そのためハイパーパラメータの調整には一般に多くの試行が必要になってしまい，計算コストが課題となる．運動制御による移動課題においては常に動的移動を可能とする制御入力を探索する必要があり，停止という局所解に陥らずにエネルギー効率の良い歩行を学習する方法が求められる．

そこで，スパイキングニューラルネットワー

ク (SNN) と呼ばれるニューラルネットワーク (NN) に着目した．SNN は脳の神経細胞をモデル化したものであり，スパイク信号によって情報を伝達する．SNN は従来の NN よりも実際の生物のニューロンに近いモデルであり，時空間情報のパターン認識に優れている．また，生物においても，ニューロンや細胞といった興奮性のシステムにおいて，ノイズが秩序の生成を誘導するということが知られており [5]，SNN に代表される，不連続なポテンシャルを持つ NN は従来の NN よりも高性能に働くことが期待されている [6] [7]．これまでスパイクは 2 値で、アナログ値を伝送する必要がないため SNN は効率の良い計算が可能 [8] という点が注目されていたが，近年では歩行運動において即座に環境に適応した動きを SNN によって探索できたということが報告されており，SNN の探索性能の高さが注目されている [9].

本研究では SNN を用いることでエネルギー効率の良い歩行を学習することを目的とする．そのため，6 脚エージェントの運動のペナルティを変化させて歩行実験を行う．DRL と SNN を組み合わせることで，従来の DRL よりも運動のペナルティが大きくても最適な運動を探索し，エネルギー効率の良い行動パターンが得られるかを検証する．

2. 問題設定

2.1 深層強化学習

強化学習ではマルコフ決定過程 $(\mathcal{S}, \mathcal{A}, p, r)$ で表される環境を考える．ここで $\mathcal{S}$ は状態空間 (今回は連続値)，$\mathcal{A}$ は行動空間を表し，$p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ は状態遷移確率関数, $r : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$ は環境との相互作用によって得られる報酬を示す．また，行動の確率分布 $\pi(a_t|s_t)$ を方策とし，これによって得られた軌跡を ρ_π として表す．本研究ではモデルフリー強化学習で，特に連続値制御のタスクにおいて広く用いられている SAC, TD3, DDPG を用いた．

SAC は確率的方策を用いる DRL アルゴリズムであり，方策のエントロピー項 $\mathcal{H}$ を考慮した目的関数 (1) を最大化する方策 $\pi(a_t|s_t)$ を学習する．

$$J(\pi) = \sum_{t=0}^{\infty} E_{(s_t,a_t)\sim\rho_\pi} \left[r\left(s_t, a_t\right) + \gamma \cdot H\left(\pi\left(\cdot \mid s_t\right)\right)\right] \tag{1}$$

エントロピー項を考慮することで，行動の多様性を保ちつつ，得られる報酬を最大化することができる．またオフポリシーでの学習が可能なことからサンプル効率に優れている．

TD3, DDPG は決定的方策を用いる DRL アルゴリズムであり，方策が $\pi(a_t|s_t) = \mu_\theta(s)$ と表され，現在の状態に対し，最適な行動を出力する．目的関数は (2) で表される．

$$J(\pi) = \sum_{t=0}^{\infty} E_{(s_t,a_t)\sim\rho_\pi} \left[r\left(s_t, a_t\right)\right] \tag{2}$$

TD3 は DDPG の探索能力と価値の過大評価を改善したアルゴリズムである．DDPG では価値の過大評価が起こってしまうが，TD3 では Q 関数を 2 つ用いる Clipped Double Q-learning という手法により価値の過大評価を改善している．また，局所的な外れ値に対し安定して学習を行うために, action に平均が 0 のガウシアンノイズを加えるという Target Policy Smoothing という手法が採用されている．

2.2 スパイキングニューラルネットワーク

スパイキングニューラルネットワーク (SNN) は (3) で表されるスパイクトレインによって情報伝達を行う，実際のスパイクニューロンの特徴を捉えたモデルであり，これまで様々なモデ

ルが提案されてきた [10] [11] [12].

$$S(t) = \sum_{s} \delta(t - t^s) \tag{3}$$

ここで s はスパイクのラベルを示し，δ はディラックのデルタ関数を示す．

SNN の中でも計算の容易さから Leaky Integrate-And-Fire(LIF) Model [13] が現在広く用いられている．LIF モデルは (4) で表される．

$$\tau \frac{du(t)}{dt} = -u(t) + I(t) \tag{4}$$

ここで $u(t)$ は時間 t におけるニューロンの膜電位であり，τ は時定数，$I(t)$ はシナプス前細胞のスパイクトレインによって決まる入力を示す．膜電位 u が閾値 V_{th} を超えたとき，ニューロンは発火し，電位は u_{reset} にリセットされる．

SNN は従来の NN に比べ，時空間パターンを持った情報のパターン認識に優れているという一方で，多層 SNN の学習において NN の誤差逆伝搬法が適応できないということが知られている．そこで，本研究では SNN の誤差逆伝搬法として高い性能を示す，Spatio-Temporal Backpropagation(STBP) 法 [14] を用いた．

3. 方法

3.1 シミュレーションで使用した 6 脚エージェント

連続値制御の強化学習において広く用いられているシミュレーションエンジンである MuJoCo [15] を用いて実験を行った．複雑な環境下において利用することを想定し，脚型ロボットを使用した．使用した 6 脚エージェント (Fig. 1) は dm_control [16] 環境上にて作成した．エージェントは 6 本の脚を持ち，各脚について肩が 2 自由度，肘が 1 自由度，手首が 1 自由度 (パッシブ) の 4 自由度, 1 つの腱をもつ．またそれぞれ肩と肘に計 3 つのアクチュエータを有する．エージェントは状態としてヒンジの位置と速度，アクチュエータの出力，胴体の速度，胴体の直立度 (胴体の z 軸と絶対座標の z 軸の内積), IMU センサーの値, そして足先にかかる力とトルクの合計 112 次元を持つ．入力 (action) は各脚のアクチュエータのトルク入力とする．

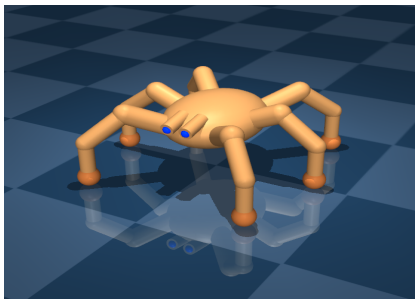


Fig. 1: 実験で使用した 6 脚エージェント

報酬関数 R を (5) と設定し，各アルゴリズムを用いて学習を行った．各 timestep において報酬関数によって報酬が与えられる．

$$R = v + s - \alpha \sum_{i=1}^{N_a} a_i(t)^2 \tag{5}$$

ここで N_a は action の個数 (ここではアクチュエータの数), v は胴体の速さ，s は生存報酬を示し，転倒していなければ timestep ごとに 1 が与えられる．a_i は action の大きさ (アクチュエータへのトルク入力の大きさ), α は action の 2 乗和に対する係数を示す．α が大きくなるにつれて，報酬に対する運動のペナルティが大きくなり，エージェントはエネルギー効率の良い行動パターンを獲得することができると考えられる．

3.2 歩行実験

エージェントに対し, 各 DRL アルゴリズムと，それらに SNN を適用したアルゴリズムを用いて学習を行った．各 DRL アルゴリズムに SNN を適応する方法として PoPSAN [17] を選択した．PoPSAN では，Population coding を通じてエージェントの状態をスパイクトレインに変

換し，STBP を拡張した extended STBP [18] によって学習を行う．Population coding はニューロンの集団の活動によって情報を表現するという方法であり，生物においても同様の応答が確認されている [19] [20].

DRL アルゴリズムは深層強化学習ライブラリである pfrl*を利用した．PoPSAN は著者らの実装†を参考にし，pfrl の DRL と組み合わせて実装を行った．Actor(NN, SNN) と Critic(NN) はそれぞれ 256 個のニューロンを 2 層組み合わせたものを使用した．

3.3 Cost of Transport

エネルギー効率の評価として Cost of Transport(CoT) を (6) に定義することで学習済みのエージェントのエネルギー効率を評価した．

$$CoT = \frac{\sum_{i=0} \int_0^t |a_i(t)\dot{\theta}_i(t)|dt}{mg\Delta d} \qquad (6)$$

ここで分子はエージェントの消費エネルギーを示し，それぞれ $a_i(t)$, $\dot{\theta}_i(t)$ は関節への入力と角速度を示す．各 Δd はエージェントの移動距離を示す．CoT は単位距離を移動するのに必要なエネルギー量を示し，小さいほどエネルギー効率の良い歩行を行っていることを示す．

4. 結果

よりエネルギー効率の良い歩行を獲得するために，DRL アルゴリズムの SAC,TD3,DDPG と，それらに SNN を適用したものを用いて，action の重み係数 α の値をそれぞれ 0 から 1 まで変化させてエージェントの学習を行った．その後，学習で得られた歩行のエネルギー効率を評価するために，CoT の計測を行った．

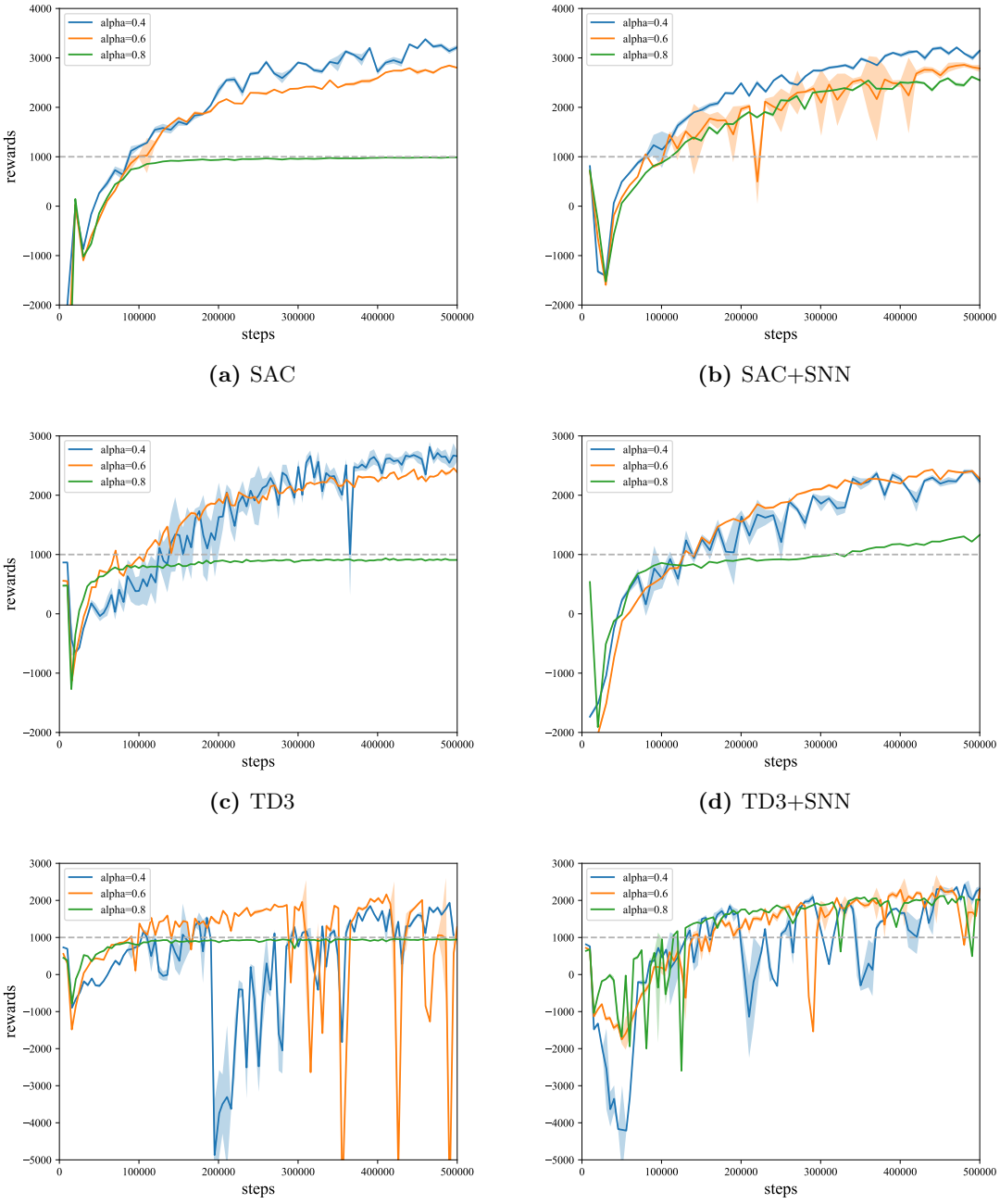
Table 1: 各アルゴリズムから得られた Cost of Transport (CoT)

		α	0.0	0.2	0.4	0.6	0.8	1.0
SAC	200k step	CoT	3.123	3.487	**2.768**	**2.099**	×	×
		SD	0.059	0.024	0.043	0.028		
	500k step	CoT	3.320	**2.796**	**2.425**	**1.913**	×	×
		SD	0.042	0.027	0.017	0.039		
TD3	200k step	CoT	7.522	3.419	**2.678**	**2.406**	×	×
		SD	0.477	0.288	0.034	0.021		
	500k step	CoT	3.621	3.766	**2.942**	**2.378**	×	×
		SD	0.151	0.115	0.067	0.021		
DDPG	200k step	CoT	12.240	5.475	17.395	**2.591**	×	×
		SD	1.161	1.458	11.426	0.085		
	500k step	CoT	5.556	3.887	**2.709**	**2.397**	×	×
		SD	0.896	0.079	0.043	0.028		
SAC+SNN	200k step	CoT	3.916	3.752	**2.586**	**2.159**	**2.354**	×
		SD	0.123	0.060	0.030	0.058	0.023	
	500k step	CoT	3.483	3.421	**2.415**	**1.900**	**1.640**	×
		SD	0.560	0.024	0.035	0.035	0.017	
TD3+SNN	200k step	CoT	4.875	3.964	**2.761**	**2.268**	×	×
		SD	0.832	0.137	0.080	0.015		
	500k step	CoT	4.339	3.588	**2.951**	**2.207**	**2.722**	×
		SD	0.127	0.139	0.064	0.014	0.013	
DDPG+SNN	200k step	CoT	9.665	4.439	3.233	**2.699**	**2.175**	×
		SD	0.836	0.250	0.375	0.707	0.012	
	500k step	CoT	3.948	**2.994**	**2.801**	**2.259**	**2.077**	×
		SD	0.960	0.088	0.113	0.034	0.019	

歩行実験の学習の推移を Fig. 2 に示す．1000 timestep を 1 episode とし，10000 timestep ごとに探索を行わない evaluation を 10 回行いながら，合計 500000 timestep のトレーニングを行った．ここで，各アルゴリズムについて，α の値が大きくなると報酬が 1000 で収束しているものがある．これはエージェントが歩行せずに停止していることを示している．

次に，学習を 200000 timestep と 500000 timestep 行ったエージェントを用いて CoT を計測した．1000step の歩行実験を 30 回行い，CoT とその標準偏差を計算した．1000step の歩行を 30 回行い，そこで得られた CoT の標準偏差を SD に示す．赤字は CoT が 2.5 未満を示し，太字は 2.5 以上 3.0 未満を示す．Fig. 1 より，ほとんどの場合，学習ステップが進むにつれて CoT が小さくなった．また α が大きくなるにつれて，より CoT の低いエネルギー効率の良い歩行を学習することができた．

*https://github.com/pfnet/pfrl
†https://github.com/combra-lab/pop-spiking-deep-rl



(a) SAC　　**(b)** SAC+SNN

(c) TD3　　**(d)** TD3+SNN

(e) DDPG　　**(f)** DDPG+SNN

Fig. 2: 各アルゴリズムにおいてそれぞれ入力の重み係数 α を変えたときの報酬の推移．１回の歩行は 1000timestep であり，報酬が 1000 で収束しているのは歩行を学習できずに停止していることを示す．

5. 考察

Fig. 2 に示すように，SAC, TD3, DDPG のすべてのアルゴリズムにおいて SNN を用いることで，$\alpha = 0.8$ において，停止という局所解に陥ることなく歩行を学習することができた．また同じ α の値においても SNN を用いていないものに比べ，低い CoT を得ることができた．これにより，CoT の低い，エネルギー効率の良い行動パターンを獲得することができた．SAC に SNN を用いた場合に一番低い CoT の歩行を得られた．SAC は他の決定的方策のアルゴリズムよりも探索能力が高いことが知られており，また SAC は TD3 と比べエネルギー効率の良い運動が得られるということが報告されている [21]．よって，SNN を用いた今回の場合においても一番エネルギー効率の良い歩行が得られたのは妥当であると考えることができる．

また，Fig. 2 によると，DDPG においては SNN を使用していないものと比べ，ばらつきの少ない歩行パターンが得られ，報酬の増加傾向も確認できた．DDPG は決定的方策により学習を行うため，計算が早いというメリットがあるものの，SAC のようなアルゴリズムと比較すると局所解に陥りやすい．そこで，今回モデルに SNN を採用することでモデル自体の探索能力が上がり，歩行において定常的な重心移動のパターンを探索することができた．それが結果として報酬の増加に寄与したと考えられる．また，DDPG においては学習後期において急激に報酬が落ちることがあったが，SNN を用いることで安定した歩行パターンを獲得したため，それが改善したと考えられる．一方で TD3 は DDPG ほど顕著な性能向上が見られなかった．TD3 では，Target policy smoothing によって価値関数の推定を行う Critic を平滑化し，クリップされたノイズを加えることで局所解に陥りにくくするという工夫がされているが，SNN が何らかの干渉を起こした可能性が考えられる．局所解に陥るのを防ぐために NN にノイズを加えるという手法が取られることがあるが，このノイズの大きさはハイパーパラメータであるため十分な効果を得るためにはタスクごとに調整する必要がある．しかし SNN によるノイズ効果はハイパーパラメータの調整が必要ないため，あらゆる深層強化学習に汎用的に利用できる可能性がある．

6. 結言

本研究では 6 脚エージェントに対し，各 DRL アルゴリズムとそれらに SNN を用いたアルゴリズムによって, action の重み係数を変えながら歩行の学習を行い，Cost of Transport(CoT) を用いてエネルギー効率の評価を行った．SAC のような確率的方策のアルゴリズムと，TD3 や DDPG のような決定的方策のアルゴリズムの両アルゴリズムにおいて，SNN を用いることでより大きな action の重み係数において歩行パターンを探索することに成功し，エネルギー効率の良い歩行を得られた．また TD3 や DDPG においては報酬の増加も確認することができた．今まで，SNN は主に省電力というメリットに着目されていたが，エネルギー効率の良い運動学習の探索という観点においても SNN が有益であるということを実験的に示すことができた．一方で，今回のモデルの探索能力向上という結果は理論的には不明瞭なところが多いため，今後は理論面においても今回の結果を考察する必要がある．

参考文献

1) T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” 2019.

2) T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *International Conference on Machine Learning (ICML)*, 2018.

3) S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," 2018.

4) J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng, "Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 4902–4909, Jul. 2019. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4419

5) B. Lindner, J. Garcı´ a-Ojalvo, A. Neiman, and L. Schimansky-Geier, "Effects of noise in excitable systems," *Physics Reports*, vol. 392, no. 6, pp. 321–424, 2004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0370157303004228

6) N. Kasabov and E. Capecci, "Spiking neural network methodology for modelling, classification and understanding of eeg spatio-temporal data measuring cognitive processes," *Information Sciences*, vol. 294, pp. 565–575, 2015, innovative Applications of Artificial Neural Networks in Engineering. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025514006562

7) J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in Neuroscience*, vol. 10, p. 508, 2016. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2016.00508

8) E. Z. Farsa, A. Ahmadi, M. A. Maleki, M. Gholami, and H. N. Rad, "A low-cost high-speed neuromorphic hardware based on spiking neural network," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 9, pp. 1582–1586, 2019.

9) S. Yonekura and Y. Kuniyoshi, "Spike-induced ordering: Stochastic neural spikes provide immediate adaptability to the sensorimotor system," *Proceedings of the National Academy of Sciences*, vol. 117, no. 22, pp. 12 486–12 496, 2020. [Online]. Available: https://www.pnas.org/content/117/22/12486

10) H. A. HODGKIN AL, "A quantitative description of membrane current and its application to conduction and excitation in nerve," 1952.

11) A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. homogeneous synaptic input," *Biological Cybernetics*, 2006.

12) E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

13) S. RB., "A theoretical analysis of neuronal variability," *Biophys J*, 1965.

14) Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in Neuroscience*, vol. 12, p. 331, 2018. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnins.2018.00331

15) E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 5026–5033.

16) Y. Tassa, S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, and N. Heess, "dm_control: Software and tasks for continuous control," 2020.

17) G. Tang, N. Kumar, R. Yoo, and K. P. Michmizos, "Deep reinforcement learning with population-coded spiking neural network for continuous control," in *4th Conference on Robot Learning (CoRL 2020)*, 2020, pp. 1–10.

18) G. Tang, N. Kumar, and K. P. Michmizos, "Reinforcement co-learning of deep and spiking neural networks for energy-efficient mapless navigation with neuromorphic hardware," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 1–8.

19) E. M. Maynard, N. G. Hatsopoulos, C. L. Ojakangas, B. D. Acuna, J. N. Sanes, R. A. Normann, and J. P. Donoghue, "Neuronal interactions improve cortical population coding of movement direction," *Journal of Neuroscience*, vol. 19, no. 18, pp. 8083–8093, 1999.

20) H.-J. Rauber, "Seeing multiple directions of motion—physiology and psychophysics," *Nature Neuroscience*, 2000.

21) J. Chai and M. Hayashibe, "Motor synergy development in high-performing deep reinforcement learning algorithms," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1271–1278, 2020.