計測自動制御学会東北支部 341 回研究集会 (2023.3.23)
資料番号 341-2

スパイク表現を用いた深層強化学習により生成された四脚ロボットの歩容評価

Evaluation of Gait of a Quadruped Robot Generated by Deep Reinforcement Learning with Spike Representation

○瀬戸崚生，沓澤京，大脇大，林部充宏

○ Ryosei Seto, 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), 四脚ロボット (quadruped)

連絡先: 〒 980-8579 宮城県仙台市青葉区荒巻字青葉 6-6-01 東北大学 青葉山キャンパス 機械系共同棟 503 瀬戸崚生 Tel.:022-795-6970 Fax.:022-795-6971 E-mail: ryosei.seto.t8@dc.tohoku.ac.jp

1. 緒言

脚ロボットはその走破性の高さから実用的かつ将来性の高い移動ロボットとして研究・開発が進められている．脚ロボットは接地面積が小さいことから不整地環境での移動に有利である．さらに，陸上を走行する生物において移動速度が速いとされる動物には四足歩行のものが多い．近年では Boston Dynamics 社が開発した Spot や Unitree Robotics 社が開発した Unitree A1 などの四脚ロボットの商用販売が開始された．これらは，危険性や作業負担の伴う建設現場や災害現場などで実用化されている．しかし，四脚ロボットは車輪やキャタピラを用いたロボットと比べ，エネルギー効率が低い．移動ロボットに搭載できるバッテリーには限りがあるため，エネルギー効率の向上は１つの課題である．また，上述の建設現場や災害現場などの悪条件での使用が想定されるため，故障や環境の変化に対するロバスト性が必要である．したがって故障や環境のノイズに対するロバスト性を担保しつつエネルギー効率の良い歩行の実現が四脚ロボットの運用において重要である．また，安全性の観点から移動速度の制御や姿勢の安定性も重要である．

近年，深層強化学習を用いた脚ロボットの制御に関する手法が数多く提案されている．Xu らの研究 [1] では階層的な制御によって，手動で設計した歩容を切り替えることで四脚ロボットの多様な環境での歩行を実現している．また，Peng らの研究 [2] では深層強化学習を用いて，イヌの動作を模倣することで歩行やスピンなどの様々な動作を生成している．しかし，これらの研究ではあらかじめ歩容を設計したり，模倣学習を行うためのデータセットを用意する必要がある．さらに，Shi らの研究 [3] では CPG (central

pattern generator) と深層強化学習を組み合わせることによって四脚ロボットの歩容を生成し，様々な地形での歩行に成功している．しかし，実際のロボットではセンサーのノイズが大きすぎるため，実機で速度センサーの値を用いた制御が困難である．一方，近年ノイズやパラメータの変化に対してのロバスト性からスパイキングニューラルネットワーク (SNN) が注目を集めている．通常の人工ニューラルネットワーク (ANN) では実数値を伝達情報としているのに対して，SNN ではニューロンの発火によって発生するスパイクの有無と発生頻度によって情報の伝達を行う．Yu らの研究 [4] ではノイズの多い感覚入力に対してロバストであることが示された．また，Lele らの研究 [5] では，SNN のフレームワークで CPG の動機パターンを学習することで，6 脚ロボットの歩行の自律的な強化学習を実現した．さらに，納谷氏らの研究 [6](Fig. 1) では，深層強化学習に SNN を適応することでシミュレーション上で六脚ロボットの歩容をゼロから生成することに成功している．SNN を用いた場合では用いない場合に比べ観測値にノイズを入れた際も通常の歩容を生成でき，エネルギー効率の低下も抑えられた．ただし，六脚ロボットは四脚ロボットに比べ安定であるため四脚ロボットでも同様に歩行可能であるか検証する必要がある．また，四脚ロボットの歩容をゼロから生成する研究は少なく，センサーのノイズに対するロバスト性を向上させた研究も少ない．

したがって，本研究の目的は深層強化学習と SNN を組み合わせた手法を用いて目標速度を設定し速度を制御しながらゼロから四脚ロボットの歩行の学習を行い，得られた歩行のロバスト性をエネルギー効率や安定性，移動速度の観点から評価，検証することである．

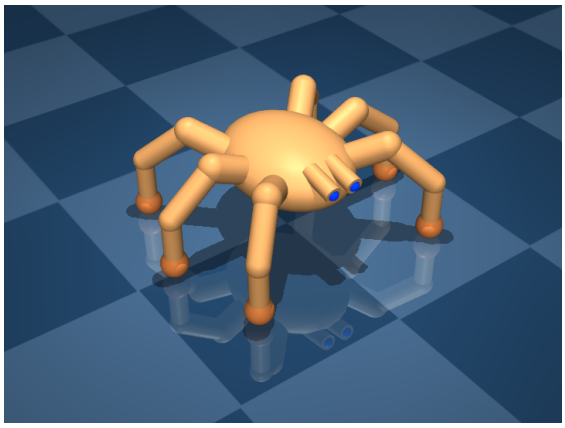


Fig. 1: Hexapod walk[6]

2. 手法

2.1 四脚エージェント

2.1.1 モデルの概要

本研究では連続値制御の強化学習で広く用いられている物理シミュレーションエンジンである MuJoCo[7] を用いて実験を行った．近年多く多く販売が開始され実用化が進んでいる四脚モデルを使用した．MuJoCo Menagerie[8] で提供されている Unitree A1 を本研究用に調整した．使用した四脚エージェントを Fig. 2 に示す．

各脚は 2 つの節を持ち，肩 (股) 関節が 2 自由度，肘 (膝) 関節が 1 自由度を持つ．よってこのエージェントは 12 自由度である．各関節には剛性が設定されており，トルク入力がない状態でも自重を支えることができ，Fig. 2 の姿勢を保持できる．エージェントの状態は Table. 1 に示すセンサ値とし，入力 (行動) は各脚 3 つのアクチュエータの角度とする．よって状態の入力が 61 次元であり行動の入力，出力は 12 次元となる．

2.1.2 報酬設計

速度報酬関数 f を式 (1) のように定義し，使用した報酬関数 r を式 (2) に示す．

$$f(v) = \begin{cases} v/v_{target} & (v \leq v_{target}) \\ v_{target}/v & (v > v_{target}) \end{cases} \tag{1}$$

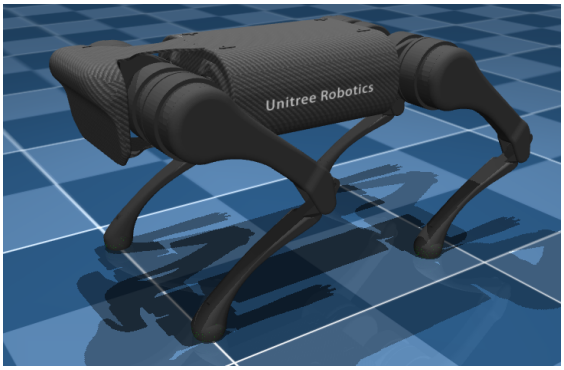


Fig. 2: Quadruped Agent

Table 1: Observations of quadruped agent

Observation	Number	Description
Trunk velocity	3	Velocity of the trunk (trunk frame)
Trunk global velocity	3	Velocity of the trunk (global frame)
Trunk gyro	3	Gyro of the trunk
Trunk acceleration	3	Acceleration of the trunk
Trunk upright	1	Inner product of the z-axis of the torso and the z-axis of the global
Actuator force	12	Force of the actuator
Actuator position	12	Position of the actuator
Actuator velocity	12	Velocity of the actuator
Foot force	12	Force on the foot
Sum	61	

$$r = f(v_{trunk}) + f(v_{global}) + \boldsymbol{z}_{global} \cdot \boldsymbol{z}_{trunk} \quad (2)$$

ここで v_{target} は設定された目標速度，v_{trunk} は胴体の座標系における胴体の前方向の速度，v_{global} はグローバル座標系における胴体の x 方向の速度である．$\boldsymbol{z}_{global} \cdot \boldsymbol{z}_{trunk}$ は地面の z 軸とエージェントの胴体の z 軸の内積であり，姿勢を安定させる役割を持つ．

2.2 評価指標

2.2.1 エネルギー効率

エネルギー効率の評価指標として移動コスト (Cost of Transport:CoT) を用いた．CoT は式 (3) で表される．

$$\mathrm{CoT} = \frac{\sum_{i=0} \int_0^t |\tau_i(t)\dot{\theta}_i(t)|dt}{mg\Delta d} \quad (3)$$

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 $\tau_i(t)$, $\dot{\theta}_i(t)$ は各アクチュエータのトルクと角速度を示す．また，m はエージェントの質量，g は重力加速度，Δd はエージェントのグローバル座標系における x 方向の移動距離を示す．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．

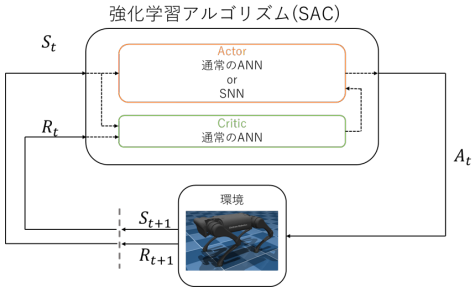


Fig. 3: Overview of SAC(ANN)&SAC(SNN)

2.2.2 安定性

安定性の評価指標として，地面の z 軸 $\boldsymbol{z}_{global}$ とエージェントの胴体の z 軸 $\boldsymbol{z}_{trunk}$ の内積の平均を用いる．この値を stability とし，式 (4) に示す．

$$\mathrm{stability} = \frac{1}{\Delta t} \sum_0^{\Delta t} \boldsymbol{z}_{global} \cdot \boldsymbol{z}_{trunk} \quad (4)$$

これは 1 から-1 の値をとり，胴体が傾かず安定している時ほど大きな値をとる．

2.2.3 速度

速度の評価として平均速度と式 (5) で示す速度の二乗平均平方根誤差 (Root Mean Squared Error:RMSE) を用いた．

$$\mathrm{RMSE} = \sqrt{1/\Delta t \sum_{i=0}^{\Delta t} (v_{global} - v_{target})^2} \quad (5)$$

この値が小さいほど目標速度に近い歩行が行えていることを示す．

2.3 歩行学習

Fig. 3 に歩行学習の全体図を示す．深層強化学習アルゴリズム (Soft Actor-Critic:SAC[9])(以後 SAC(ANN) と呼称) とそれらに PopSAN[10]

を用いて Actor 部の ANN を SNN に置き換えたアルゴリズム (以後 SAC(SNN) と呼称) の 2 つを用いて四脚エージェントの歩行学習を行った．目標速度は 1,2,3m/s とし，SAC(ANN) と SAC(SNN) を各速度で 3 回ずつ学習した．シミュレーションのタイムステップは 0.02 秒とし，1000 タイムステップを 1 エピソードとして合計 500000 タイムステップの学習を行った．また，転倒時にはエピソードを終了し，次のエピソードを開始した．

2.4　歩行実験

学習したパラメータを用いて 10000 タイムステップの歩行実験を学習したパラメータ 1 つにつき 3 回行った．さらに目標速度 2m/s のパラメータを用いて速度センサー (胴体の座標系のみ) と IMU(加速度センサー，ジャイロセンサー) の観測値にガウシアンノイズを加えて同様に歩行実験を行った．ガウシアンノイズは正規分布に従い，標準偏差 $\sigma = 1, 100$ の 2 パターン行った．

3.　結果

3.1　歩行学習

SAC(ANN) と SAC(SNN) を用いた学習曲線を Fig. 4 に示す．3 回の実験の標準偏差を帯で示している．学習過程においては SAC(ANN) と SAC(SNN) にはほとんど差がなかった．また，SAC(ANN),SAC(SNN) ともに目標速度が小さいほど報酬が大きくなる傾向がある．

3.2　歩行実験

3.2.1　目標速度ごとの歩行

各目標速度で歩行実験を行い，エネルギー効率，安定性，速度の評価を行った結果をそれぞれ Fig.5a-5d に示す．目標速度が大きくなるほど目標速度から離れた速度の歩容が生成された

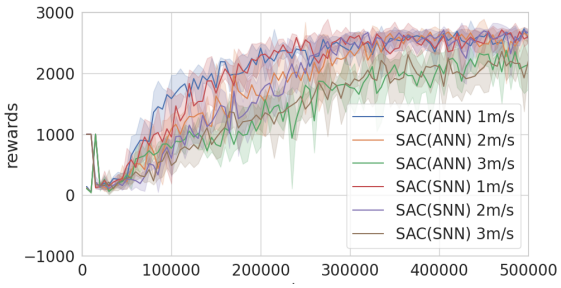


Fig. 4: Learning curve of walking of the quadruped agent

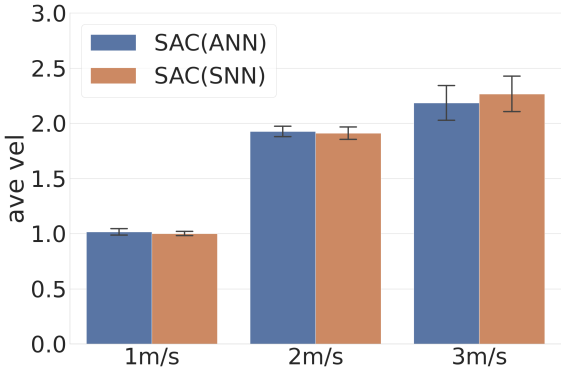
(5a,5b)．また，目標速度の増加による安定性の低下はそれほど大きくなかった (Fig. 5c)．しかし，1m/s の場合に移動コストが一番大きくなった (Fig. 5d)．

3.2.2　センサーノイズを加えた歩行

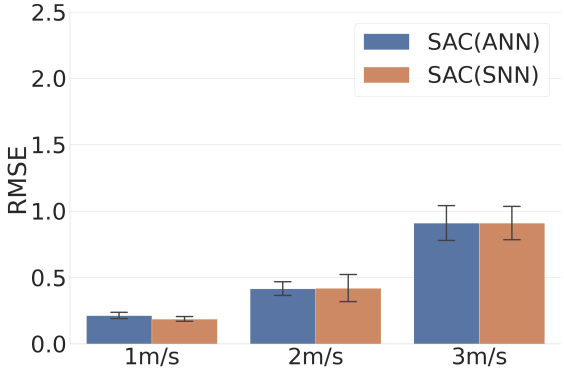
センサーの観測値にノイズを加え歩行実験を行いエネルギー効率，安定性，速度の評価を行った結果をそれぞれ Fig. 6a- 6d に示す．ノイズを加えると速度，安定性はともに低下し，移動コストは大きくなった．($\sigma = 100$ のとき SAC(ANN) では平均速度が負であったため CoT が計算できなかった．) SAC(ANN) では $\sigma = 1, 100$ のどちらの場合も通常の歩行が困難であったのに対し，SAC(SNN) では $\sigma = 1$ のとき通常時に近い歩行が行えた．

4.　考察

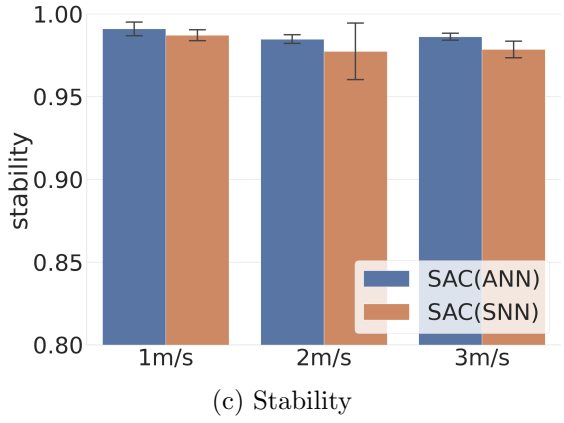
速度報酬関数を設定したことにより不用意に速度の大きすぎる歩容が生まれることはなかった．1,2m/s の歩行においては目標速度に近い速度での歩行が生成できたが 3m/s では目標速度の 7 割程度の速度になってしまったことで，速度の目標速度との誤差が大きくなってしまった．(Fig.5a,5b) これは，今回設定した速度報酬が目標速度が高いほど速度の増加に伴う報酬の増加量が小さいため学習過程において速度報酬よりも地面の z 軸とエージェントの胴体の z 軸の内積による安定性の報酬を優先させたからである



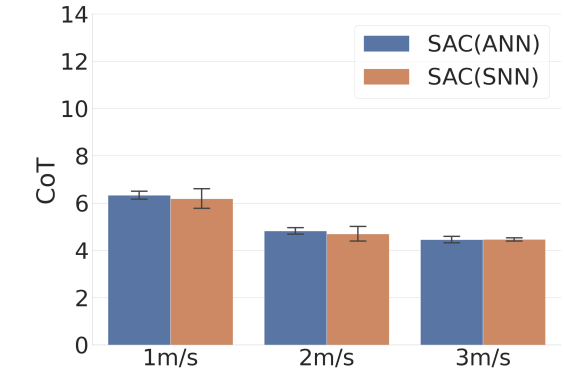
(a) Average Velocity



(b) RMSE of Velocity

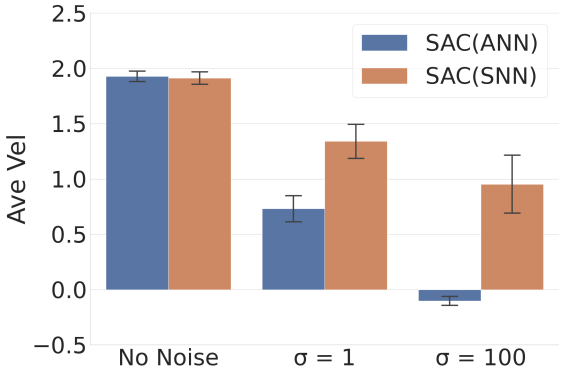


(c) Stability

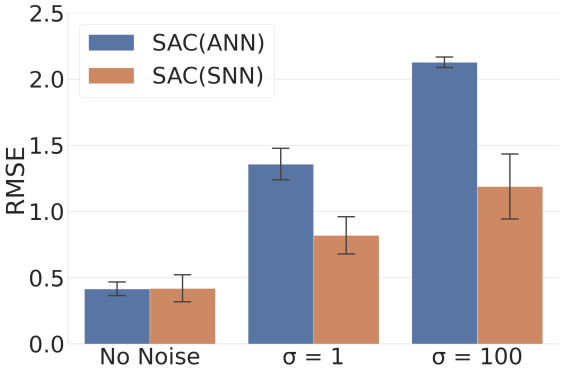


(d) CoT

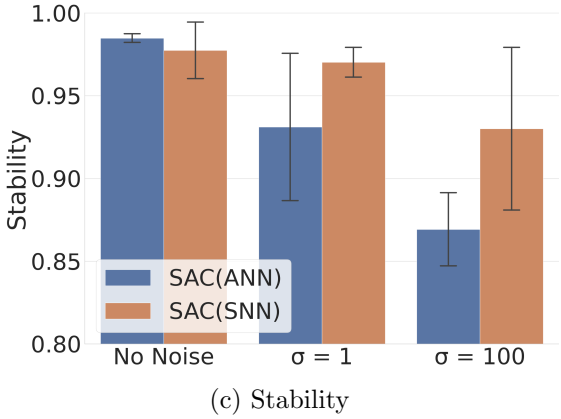
Fig. 5: Evaluation of quadruped walk (target velocity)



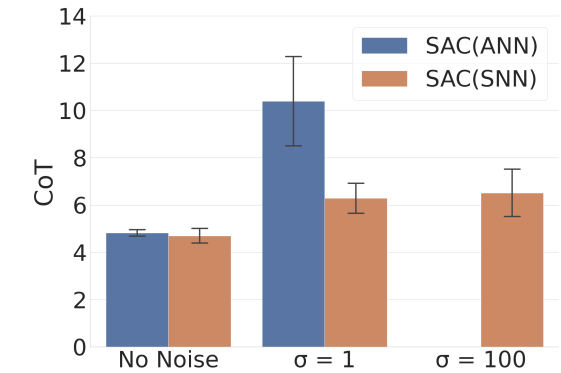
(a) Average Velocity



(b) RMSE of Velocity



(c) Stability



(d) CoT

Fig. 6: Evaluation of quadruped walk (Noise)

と考えられる．また，移動コストが目標速度が1m/s の時一番大きくなった要因としては速度を抑えるために小さく跳ねる歩容を学習してしまったためであると考えられる (Fig.5d).

センサーノイズを加えた実験では，速度センサーや加速度センサー，ジャイロセンサーの観測値にノイズを入れると移動コストの上昇や安定性や速度の低下がみられることがわかった．SAC(SNN) ではノイズによる移動コストの上昇が抑えられた (Fig6d)．その要因として Fig.6a から分かるように SAC(SNN) では SAC(ANN) よりも速度の低下が小さかったことや Fig.6c から見て取れるように安定性の低下が抑えられたことが挙げられる．よって SNN を用いたことでセンサーノイズに対してロバストになったと言える．SNN がノイズに対してロバストであることはスパイク表現を用いて情報伝達を行うためだと考えられている [11]．

5. 結論

本研究では深層強化学習とスパイキングニューラルネットワーク (SNN) を組み合わせて目標速度を設定し四脚ロボットの歩行学習を行った．その後，センサーノイズを加えた歩行実験を行い，エネルギー効率や安定性，移動速度などの観点から歩行の評価を行った．

本研究では深層強化学習と SNN を組み合わせたアルゴリズムを使用して，シミュレーション上で四脚ロボットの歩容生成に成功した．また，速度報酬関数を設定したことにより速度の大きすぎる歩容が生成されることはなく，速度面における安全性が担保できた．さらに，SNN を用いたことでセンサーノイズに対するロバスト性が向上し，ノイズによるエネルギー効率や安定性，移動速度の低下を抑えることができた．

今後の展望としてより自然な歩容の生成を挙げる．本研究では速度報酬と安定性の報酬の二つを報酬として用いたが，さらに報酬の設計を工夫することでより自然な歩容を生成できる可能性がある．また，本研究で示された SNN のセンサーノイズに対するロバスト性を実機でも検証することが求められる．

参考文献

1) Shaohang Xu, Lijun Zhu, and Chin Pang Ho. Learning efficient and robust multi-modal quadruped locomotion: A hierarchical approach. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 4649–4655, 2022.

2) Xue Bin Peng, Erwin Coumans, Tingnan Zhang, Tsang-Wei Edward Lee, Jie Tan, and Sergey Levine. Learning agile robotic locomotion skills by imitating animals. In *Robotics: Science and Systems*, 07 2020.

3) Haojie Shi, Bo Zhou, Hongsheng Zeng, Fan Wang, Yueqiang Dong, Jiangyong Li, Kang Wang, Hao Tian, and Max Q.-H. Meng. Reinforcement learning with evolutionary trajectory generator: A general approach for quadrupedal locomotion. *IEEE Robotics and Automation Letters*, Vol. 7, No. 2, pp. 3085–3092, 2022.

4) Qiang Yu, Rui Yan, Huajin Tang, Kay Chen Tan, and Haizhou Li. A spiking neural network system for robust sequence recognition. *IEEE Transactions on Neural Networks*, 2016.

5) Ashwin Sanjay Lele, Yan Fang, Justin Ting, and Arijit Raychowdhury. Learning to walk: Spike based reinforcement learning for hexapod robot central pattern generation. In *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 208–212, 2020.

6) Katsumi Naya, Kyo Kutsuzawa, Dai Owaki, and Mitsuhiro Hayashibe. Spiking neural network discovers energy-efficient hexapod motion in deep reinforcement learning. *IEEE Access*, Vol. 9, pp. 150345–150354, 2021.

7) 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*, pp. 5026–5033, 2012.

8) MuJoCo Menagerie Contributors. MuJoCo Menagerie: A collection of high-quality simulation models for MuJoCo, 9 2022.

9) Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International Conference on Machine Learning (ICML)*, 2018.

10) Guangzhi Tang, Neelesh Kumar, Raymond Yoo, and Konstantinos P Michmizos. Deep reinforcement learning with population-coded spiking neural network for continuous control. In *4th Conference on Robot Learning (CoRL 2020)*, pp. 1–10, 2020.

11) Chen Li, Runze Chen, Christoforos Moutafis, and Steve Furber. Robustness to noisy synaptic weights in spiking neural networks. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, United States, September 2020. IEEE.