Hybrid Spiking Neural Networks for Low-Power Intra-Cortical Brain-Machine Interfaces

Alexandru Vasilache*,[1,2] Jann Krausse*,[1,3]
Klaus Knobloch,[3] Juergen Becker [1]
[1] *Karlsruhe Institute of Technology, Karlsruhe, Germany*
[2] *FZI Research Center for Information Technology, Karlsruhe, Germany*
[3] *Infineon Technologies, Dresden, Germany*

***Abstract*—Intra-cortical brain-machine interfaces (iBMIs) have the potential to dramatically improve the lives of people with paraplegia by restoring their ability to perform daily activities. However, current iBMIs suffer from scalability and mobility limitations due to bulky hardware and wiring. Wireless iBMIs offer a solution but are constrained by a limited data rate. To overcome this challenge, we are investigating hybrid spiking neural networks for embedded neural decoding in wireless iBMIs. The networks consist of a temporal convolution-based compression followed by recurrent processing and a final interpolation back to the original sequence length. As recurrent units, we explore gated recurrent units (GRUs), leaky integrate-and-fire (LIF) neurons, and a combination of both - spiking GRUs (sGRUs) and analyze their differences in terms of accuracy, footprint, and activation sparsity. To that end, we train decoders on the "Nonhuman Primate Reaching with Multichannel Sensorimotor Cortex Electrophysiology" dataset and evaluate it using the NeuroBench framework, targeting both tracks of the IEEE BioCAS Grand Challenge on Neural Decoding. Our approach achieves high accuracy in predicting velocities of primate reaching movements from multichannel primary motor cortex recordings while maintaining a low number of synaptic operations, surpassing the current baseline models in the NeuroBench framework. This work highlights the potential of hybrid neural networks to facilitate wireless iBMIs with high decoding precision and a substantial increase in the number of monitored neurons, paving the way toward more advanced neuroprosthetic technologies.**

***Index Terms*—spiking neural network, neural decoding, brain machine interface, neurobench**

I. Introduction

Tens of millions of lives worldwide are suffering from paralysis [1], [2]. Those affected experience an impaired ability to direct their movements, which, in severe cases, leads to a complete loss of motor control. This motivates the development of technology that can decode patients' brain activity and accordingly control assistive prostheses. Such devices are called brain machine interfaces (BMIs) [3] and have been very successful with restoring motor control [4], sensory information [5], or even emotional responses [4].

Usually, BMIs are directly placed on the surface of a patient's brain to ensure the maximal quality of the recorded brain signals (iBMIs). However, this raises two problems. First, implants are connected via bulky wiring to the operating equipment, severely restricting the patient's movement [6]. Second, permanently opening the skull to allow wiring increases the risk of infection [7]. In hopes of mitigating this, research is moving towards wireless iBMIs [6], [8].

The Grand Challenge on Neural Decoding for Motor Control of non-Human Primates of IEEE BioCAS 2024 calls for solutions to the scalability issues of such wireless BMIs. Since data rates are limited due to bit-error rates, heat dissipation, and battery lifetime, an optimal solution should handle the trade-off between high-quality neural decoding, data compression, and resource management. As the development of techniques for embedded artificial intelligence progresses, neural networks present promising candidates for wireless low-power neural decoders [9], [10]. Additionally, biologically inspired spiking neural networks (SNNs) benefit from high temporal sparsity, single-bit communication facilitated by spikes, and an intrinsic recurrence due to their statefulness [11]. Consequently, participants of the Grand Challenge on Neural Decoding are tasked with training a neural network on the Primate Reaching dataset [12] for predicting the velocities of cursor movements. The network is then evaluated using the NeuroBench framework to obtain metrics regarding accuracy and resources [13]. Results are judged based on two challenge tracks: track 1 assesses sole accuracy optimization, while track 2 targets the co-optimization of accuracy, memory footprint, and number of compute operations, as defined in [13].

Our work presents a hybrid network architecture of temporal convolutions in combination with recurrent processing and a subsequent interpolation back to the original sequence length. While GRUs are very effective in sequence modeling [14], networks based on spiking neurons like the LIF model profit from the advantages of SNNs regarding resourcefulness mentioned above [15]. Hence, we investigate recurrent processing by GRUs, LIF units, and a combination of both and discuss the differences in their results.

Furthermore, we motivate the chosen architecture via a few experiments before presenting the results of all three types of recurrence. All three network types beat the baselines given by [13] in at least one of the challenge tracks by a good margin. However, the different recurrence types show evident differences in accuracy and resourcefulness. Based

The project on which this report is based was sponsored by the German Federal Ministry of Education and Research under grant number 16ME0801. The responsibility for the content of this publication lies with the author.

*These authors contributed equally to this work.

on that, we will discuss the implications of using spiking elements. Finally, we point out the possibilities of the real-time deployment of these networks and areas of future work.

II. Related Work

The authors of [16] used SNNs to predict a rhesus monkey's arm velocity accurately. However, the network was not trained directly on the data. Instead, they mapped a Kalman filter onto the network.

In [17], the authors train SNNs on two datasets for offline finger velocity decodings. They achieve high accuracy and compare their approach to the artificial neural networks (ANNs) baseline, even specifying numbers for total operations and memory accesses. Still, their network represents a simple feed-forward architecture and is trained on a different dataset than this work.

The clear baseline for this work is given by [13]. Among other datasets, the authors make the dataset of [12] available for deep learning approaches and subsequently train neural networks as baselines. They differentiate between ANNs and SNNs as well as between networks that target pure reconstruction accuracy (track 1) and those that co-optimize accuracy and resource demands (track 2). The used networks are of relatively simple architecture. Their work aims to enable others to benchmark respective datasets easily. We will make use of their work and surpass their baseline using a different network architecture in both challenge tracks.

III. Methods

A. Motivating an Interpolation-based Approach

Our interpolation approach is inspired by observing primate cursor movements. In the video, a new target appears each time the previous one is reached, prompting a rapid, goal-directed movement toward it. This suggests that the movement can be approximated by discrete, target-locked actions rather than fine-grained continuous adjustments.

Based on this, we hypothesize that capturing a few keypoints along the velocity trajectory and interpolating between them can effectively approximate the whole movement velocity. Fig. 1 illustrates this concept by comparing the original movement with a simplified version, where the velocity at every eighth point is retained, and linear interpolation is used between them. We argue that the resulting error is negligible, assuming that the keypoint prediction is of high quality, as the R2 score between the interpolated test set and the original test set is 0.998 with 4-step interpolation, 0.988 with 8-step interpolation and 0.955 with 16-step interpolation.

B. Model Architecture

The general architecture of the model (Fig. 2) involves temporal convolutions to reduce the number of time steps in a sequence of neuron recordings from the input size of 1024 to the desired number of keypoints and efficiently extract temporal features. To create sufficient keypoint pairs, convolutional blocks reduce the sequence to a length of *number of keypoints + 1*. These features are then processed by recurrent units and a fully connected layer to determine output velocities as keypoints. We apply linear interpolation between the determined keypoints to scale the output sequence back to the original sequence length.

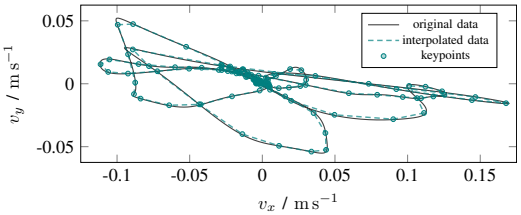

Fig. 1: Linear interpolation of discretized cursor velocities (8 steps) visualized above original cursor velocities.

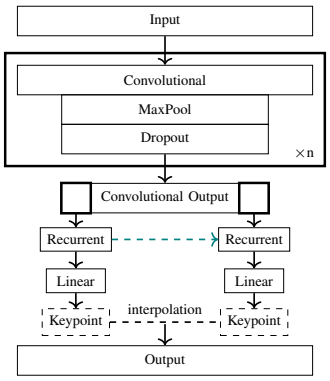

Fig. 2: Illustration of the general network architecture used in this work.

Here, we compare three types of recurrent units for the architecture described above. Those comprise GRU and LIF units, as well as a fusion of both, which we call the sGRU. We define the sGRU as

$$\mathbf{r}_t = \text{LIF}(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1}), \tag{1}$$

$$\mathbf{z}_t = \text{LIF}(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1}), \tag{2}$$

$$\tilde{\mathbf{h}}_t = \text{LIF}(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h((1 - \mathbf{r}_t) \odot \mathbf{h}_{t-1})), \tag{3}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \tilde{\mathbf{h}}_t, \tag{4}$$

where $\mathbf{r}_t$, $\mathbf{z}_t$, $\tilde{\mathbf{h}}_t$, and $\mathbf{h}_t$ denote reset gate, update gate, candidate hidden state, and hidden state at time t, respectively. $\mathbf{W}_r$, $\mathbf{W}_z$, $\mathbf{W}_h$, $\mathbf{U}_r$, $\mathbf{U}_z$, and $\mathbf{U}_h$ are learnable parameters. $\mathbf{x}_t$ denotes the input. LIF refers to the implementation of the LIF spiking neuron model presented in [15].

Fig. 3 displays the intermediary states of all three network types for visualization. Based on this general architecture, we present 2 model sizes, targeting track 1 (GRU-t1, sGRU-t1, LIF-t1) and track 2 (GRU-t2, sGRU-t2, LIF-t2) of the Neural Decoding Challenge. The LIF networks additionally use recurrent weights. Track 1 models employ three convolutional blocks with 32 channels, kernel sizes of 3, 6, and 12, and padding sizes of 5, 3, and 6, targeting 8-step interpolation with 127 keypoints. All max pooling layers use a kernel size and stride of 2. The size of the recurrent blocks is 64. Track 2 models use two convolutional blocks with 10 channels and a kernel size of 3, which reduce the

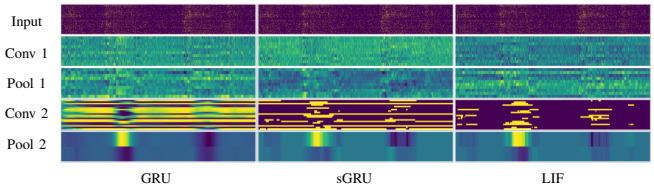

Fig. 3: Layer visualisations for GRU-t2, sGRU-t2, LIF-t2.

Table I: Tradeoff between model size and R2 Score

Conv Channels x GRU Hidden Size	R2 Score
10x20	0.667
32x64	0.692
64x128	0.687
128x256	0.661

input size to 257 keypoints, effectively targeting a 4-step interpolation. To achieve the number of keypoints, the first convolutional layer uses a padding of 3, while the second convolutional layer employs a padding size of 1. The max pooling layers both use a kernel size and stride of 2. The size of the recurrent blocks is 20.

IV. Experiments and Observations

To understand the relationship between model size and the R2 score and the tradeoff that comes with it, we trained four networks of different hidden sizes. Due to time limitations, we performed this experiment only for the sGRU model, training only on the indy2016062201 file with fewer data samples. Table I displays the respective results.

Additionally, we study the influence of the number of keypoints on the R2 score by training four networks with 1025 to 129 keypoints (1-step to 8-step interpolation). Again, we trained the networks only on the indy2016062201 file with fewer data samples. Table II presents the results for the GRU model. Note that fewer keypoints directly translate to a higher R2 score. This trend was also confirmed for sGRU- and LIF-based networks.

We also ran experiments to evaluate the test performances of models trained on all three recordings for each primate. Interestingly, the R2 score decreases when using aggregated data, contrasting the expected increase in generalizability due to a more representative training set. This hints at a possible change or degradation of the signal recording from the intracortical electrodes across time.

V. Results

A. Baseline Comparison

We present the best results we obtained for GRU-, sGRU-, and LIF-based networks for challenge tracks 1 and 2 in Table III, on the metrics defined in [13]. Comparing our models to those provided by the baselines in [13], we notice a larger footprint due to the increased input buffer size required for an input of size 1024 and the convolutional blocks. However, our models present fewer synaptic operations, judging by the Dense, MACs, and ACs values.

All our track 1 models achieve equal or higher R2 scores than the baselines, with GRU-t1 reaching an R2 score that is

Table II: Comparison of R2 scores for different number of keypoints. The networks are based on the GRU unit and do not differ in memory footprint.

Keypoints (Interpolation)	Dense Operations	Effective MACs	R2
1025 (1-step)	46400.3	37184.3	0.736
513 (2-step)	27808.3	18592.3	0.766
257 (4-step)	20054.3	10838.3	0.764
129 (8-step)	17734.3	8518.3	0.779

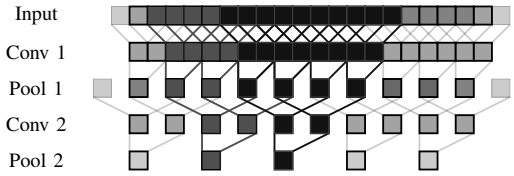

Fig. 4: Receptive Field Visualisation

increased from 0.615 to 0.707 compared to B-ANN3, while using 26% fewer MACs and 34% less Dense operations.

For track 2, we compare GRU-t2 with B-ANN2; it has only roughly 13% of the MACs and an increased R2 score ($+0.045$). sGRU-t2 uses 60% of the ACs, with the same R2 score when compared to B-SNN2 and 8% of the MACs for the same R2 score when compared to B-ANN2. The LIF-t2 model achieves the same R2 score, with roughly the same activation sparsity, while only using 63% of the Dense and 60% of the ACs when compared to B-SNN2.

B. Recurrence Comparison

By far, the best performance has been achieved by the GRU recurrent unit for both investigated sizes. Furthermore, the GRU also gives the best trade-off between footprint and R2 score. Across both sizes, the lowest number of synaptic operations (Dense and MACs) is achieved by the LIF recurrence, which reaches the highest activation sparsity, as can be visually confirmed in Fig. 3. The sGRU recurrence achieves higher activation sparsity and lower MACs than the GRU for the same number of total synaptic operations and ACs at the cost of a higher footprint and lower R2 score. Compared to the LIF, sGRU consistently displays a slightly higher R2, hinting at improved memory management, compared to solely using LIF neurons.

VI. Discussion

We hypothesize that the reason for the higher achieved R2 score, given a sufficiently large receptive field (as seen in our models proposed for track 1), may be that the filtering operations performed by the convolutional layers offer better information aggregation across time, compared to the simple summing aggregation used by the baseline model B-SNN3.

The proposed models use an input buffer window of 1024 steps provided by the NeuroBench [13] Primate Reaching Dataset, where each step represents 4 ms. This results in a total buffer window and a latency of 4.096 s. The models are executed for non-overlapping windows of size 1024, meaning that the model execution rate is 0.244 Hz.

Table III: Results of the trained networks and their respective baselines. Networks prefixed with a B refer to the baselines given by [13]. Section III-B describes the corresponding network architectures. The exact definitions of each metric are defined in [13]. The values for Dense, MACs and ACs are computed by averaging the total over the length of the input (1024), as implemented by the Neurobench benchmarking tool [13].

Track	Model	Footprint	Connection Sparsity	Activation Sparsity	Dense	MACs	ACs	R2
Track 1	B-ANN3	137752	0	0.681	33888	11507	0	0.615
	B-SNN3	**33996**	0	0.788	43680	32256	5831	0.633
	GRU-t1	352904 ± 0	0 ± 0	0 ± 0	22342 ± 0	8518 ± 0	793 ± 0	**0.707 ± 0.012**
	sGRU-t1	425924 ± 0	0 ± 0	0.651 ± 0.017	22318 ± 0	7238 ± 24	797.7 ± 0.8	0.656 ± 0.013
	LIF-t1	302492 ± 0	0 ± 0	**0.939 ± 0.008**	**20766 ± 0**	6414 ± 0	825 ± 4	0.648 ± 0.022
Track 2	B-ANN2	**27160**	0	0.676	6237	4970	0	0.576
	B-SNN2	29248	0	**0.998**	7300	0	414	0.581
	GRU-t2	174104 ± 0	0 ± 0	0 ± 0	4947 ± 0	627 ± 0	248 ± 0	**0.621 ± 0.014**
	sGRU-t2	180716 ± 0	0 ± 0	0.69 ± 0.07	4932 ± 0	379 ± 23	250.2 ± 0.8	0.577 ± 0.013
	LIF-t2	168596 ± 0	0 ± 0	0.946 ± 0.009	**4631 ± 0**	201 ± 0	254 ± 0.8	0.566 ± 0.016

Our current approach comes with a high flexibility in the possible latency and execution rate that it can achieve, as both the convolutional and the recurrent layers allow for iterative data processing. Models GRU-t2, sGRU-t2, and LIF-t2 use a kernel size of 3 applied in two convolutional blocks. The receptive field determined by this structure can be visualized in Fig. 4. With the sizes mentioned above, the computation of one keypoint requires an effective buffer window of 10 steps, which offers a latency of 40 ms. This would also reduce the input buffer size from 1024 to 10, reducing the model footprints by a sizable amount. The stride of the receptive field is 4 steps, or 16 ms, which translates to an execution rate of 62.5 Hz. The theoretical upper limit of the latency of our models (40 ms) is well under the time delay between stimulus and voluntary muscle movement reported by the neuroscience literature [18], which is typically greater than 100 ms. Assuming no further latencies arise from signal transmission and ignoring computation time, our approach would be suitable for deployment in the real world, given an appropriate real-time implementation of the networks.

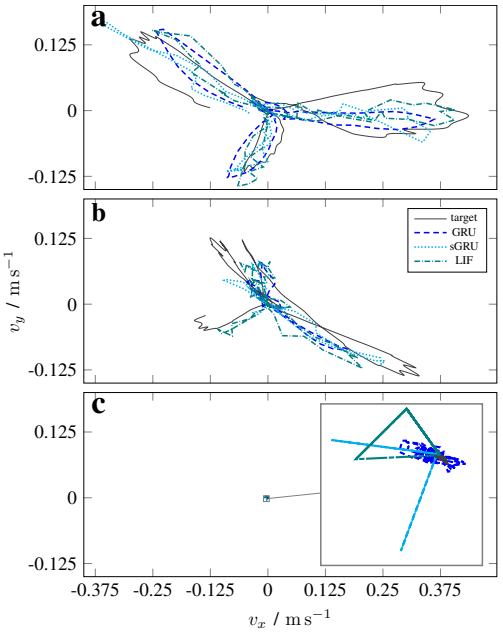

Fig. 5: Visualization of the velocity outputs of all three model types for three exemplary samples each. **a** displays a sample learned well by all three networks (R2 ≈ 0.9). **b** shows the output for a sample for which the networks display average accuracy (R2 ≈ 0.7). For the sample shown by **c**, the networks could not accurately reconstruct the target (R2 ≪ 0).

VII. Conclusion and Outlook

This work targets both tracks of the Grand Challenge on Neural Decoding for Motor Control of non-Human Primates of IEEE BioCAS 2024. This includes track 1, which focuses on maximizing task accuracy, and track 2, which aims at co-optimizing accuracy and resource demand, which is critical for wireless iBMIs. The networks presented in this work surpass the baselines in [13] by good margins for both tracks.

For track 1, GRU- and sGRU-based networks beat the baselines by up to 7.4% in terms of R2 while the LIF-based networks perform equal. For track 2, considering the margin of error, all networks are at least equal in R2 but show an improvement in the double-digit percentages in terms of compute operations. Only the footprint is increased by a rough factor of 6. We explain that this difference is due to large data buffers in our current model implementation. This gap could be eliminated in real-world deployment by taking advantage of the iterative nature of convolutional filters and recurrent units. Generally, the GRU-based networks score the highest in both tracks. However, the total amount of operations is the fewest for the LIF-based networks. Our sGRU-based models consistently achieve a higher R2 than solely using LIF-neurons, suggesting that such spiking neuron models could benefit from improved memory management.

Our work does not yet leverage some SNN-centered methods to improve their resourcefulness. This includes spike regularization, pruning, and event-triggered updating of the neural units, which will be included in future work. Finally, three of the six recordings in the dataset consist of motor cortex and somatosensory cortex recordings. We do not yet distinguish between the two different data types and expect an improved regression if done so.

Our work enhances the baseline for the primate reaching dataset and demonstrates the potential of using hybrid neural networks for efficient neural decoders. This advances the field of wireless iBMIs to eventually improve the lives of millions of humans suffering from paralysis.

REFERENCES

[1] B. S. Armour, E. A. Courtney-Long, M. H. Fox, H. Fredine, and A. Cahill, "Prevalence and causes of paralysis—united states, 2013," *American journal of public health*, vol. 106, no. 10, pp. 1855–1857, 2016.

[2] World Health Organization (WHO), "Spinal cord injury.," 2024. [Accessed: Sep. 2, 2024].

[3] M. A. Lebedev and M. A. Nicolelis, "Brain–machine interfaces: past, present and future," *TRENDS in Neurosciences*, vol. 29, no. 9, pp. 536–546, 2006.

[4] M. M. Shanechi, "Brain–machine interfaces from motor to mood," *Nature neuroscience*, vol. 22, no. 10, pp. 1554–1564, 2019.

[5] M. Lebedev, "Brain-machine interfaces: an overview," *Translational Neuroscience*, vol. 5, pp. 99–110, 2014.

[6] C. Libedinsky, R. So, Z. Xu, T. K. Kyar, D. Ho, C. Lim, L. Chan, Y. Chua, L. Yao, J. H. Cheong, *et al.*, "Independent mobility achieved through a wireless brain-machine interface," *PLoS One*, vol. 11, no. 11, p. e0165773, 2016.

[7] C. Pandarinath, P. Nuyujukian, C. H. Blabe, B. L. Sorice, J. Saab, F. R. Willett, L. R. Hochberg, K. V. Shenoy, and J. M. Henderson, "High performance communication by people with paralysis using an intracortical brain-computer interface," *elife*, vol. 6, p. e18554, 2017.

[8] J. D. Simeral, T. Hosman, J. Saab, S. N. Flesher, M. Vilela, B. Franco, J. N. Kelemen, D. M. Brandman, J. G. Ciancibello, P. G. Rezaii, *et al.*, "Home use of a percutaneous wireless intracortical brain-computer interface by individuals with tetraplegia," *IEEE Transactions on Biomedical Engineering*, vol. 68, no. 7, pp. 2313–2325, 2021.

[9] X. Zhang, Z. Ma, H. Zheng, T. Li, K. Chen, X. Wang, C. Liu, L. Xu, X. Wu, D. Lin, *et al.*, "The combination of brain-computer interfaces and artificial intelligence: applications and challenges," *Annals of translational medicine*, vol. 8, no. 11, 2020.

[10] J. Krausse, M. Neher, I. Fuerst-Walter, C. Weigelt, T. Harbaum, K. Knobloch, and J. Becker, "On metric-driven development of embedded neuromorphic ai," in *2024 IEEE 37th International System-on-Chip Conference (SOCC)*, IEEE, 2024.

[11] W. Maass, "Networks of spiking neurons: the third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[12] J. E. O'Doherty, M. M. Cardoso, J. G. Makin, and P. N. Sabes, "Nonhuman primate reaching with multichannel sensorimotor cortex electrophysiology," *Zenodo http://doi. org/10.5281/zenodo*, vol. 583331, 2017.

[13] J. Yik, S. H. Ahmed, Z. Ahmed, B. Anderson, A. G. Andreou, C. Bartolozzi, A. Basu, D. den Blanken, P. Bogdan, S. Buckley, *et al.*, "Neurobench: Advancing neuromorphic computing through collaborative, fair and representative benchmarking," 2023.

[14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[15] G. Bellec, F. Scherr, A. Subramoney, E. Hajek, D. Salaj, R. Legenstein, and W. Maass, "A solution to the learning dilemma for recurrent networks of spiking neurons," *Nature communications*, vol. 11, no. 1, p. 3625, 2020.

[16] J. Dethier, P. Nuyujukian, C. Eliasmith, T. Stewart, S. Elasaad, K. V. Shenoy, and K. A. Boahen, "A brain-machine interface operating with a real-time spiking neural network control algorithm," *Advances in neural information processing systems*, vol. 24, 2011.

[17] J. Liao, L. Widmer, X. Wang, A. Di Mauro, S. R. Nason-Tomaszewski, C. A. Chestek, L. Benini, and T. Jang, "An energy-efficient spiking neural network for finger velocity decoding for implantable brain-machine interface," in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pp. 134–137, IEEE, 2022.

[18] I. L. Kurtzer, "Long-latency reflexes account for limb biomechanics through several supraspinal pathways," *Frontiers in Integrative Neuroscience*, vol. 8, Jan. 2015. Publisher: Frontiers.