Proceedings of the DataNinja sAIOnARA 2024 Conference 53-55 DOI: 10.11576/dataninja-1170

Bioinspired Decentralized Hexapod Control with a Graph Neural Network

Luca Hermes LHERMES@TECHFAK.UNI-BIELEFELD.DE
Bielefeld University, Germany

Barbara Hammer BHAMMER@TECHFAK.UNI-BIELEFELD.DE
Bielefeld University, Germany

Malte Schilling MALTE.SCHILLING@UNI-MUENSTER.DE
University of Münster, Germany

Abstract

Legged locomotion enables animals to navigate challenging terrains. However, it demands intricate coordination between the legs, with varying levels of information exchange depending on the task. For instance, in more demanding scenarios such as an insect climbing on a twig, greater coordination between the legs is necessary to achieve adaptive behavior. To address this challenge for legged robots, we present a concept and preliminary results of a decentralized biologically inspired controller for a hexapod robot: Based on insights of coordination influences between legs in stick insects, our approach models inter-leg information flow as message passing through a Graph Neural Network.

Keywords: Reinforcement Learning, Hexapod, Decentralized Control

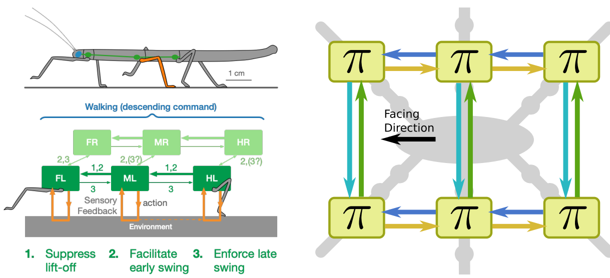
1. Introduction

Insects can traverse difficult terrain with ease while coordinating their six legs in an efficient way. This coordination manifests as a continuum of gaits that allows insects to move efficiently at different velocities. Stick insects that walk slowly exhibit the *tetrapod gait* which transitions smoothly into the *tripod gait* with increasing walking speed. Two main principles have been discovered in insect locomotion. First, insect locomotion can be modeled by a set of local rules or influences between legs [1, 2] visualized in Figure 1(*a*). Here, local means that influences exist only between immediately neighboring legs. Second, the same rules hold for each leg. Both of those principles indicate a decentralized system, where the same controller actuates every leg. This motivates an extension to the existing work [3, 4] that implements such a decentralized controller for a quadruped and a hexapod robot based on a reinforcement learning (RL) multi-agent framework. This work assigns separate neural networks to the four legs and concludes that information exchange between legs is required to facilitate functioning coordination. Here we want to go a step further by 1) using the exact same neural network to control every leg, which aligns with the second principle 2) utilize a graph neural network to implement inter-leg coordination and 3) discuss how to design a model that is transparent w.r.t. learned coordination rules.

2. Methods

We now outline a simple bioinspired model which can learn the tripod gait as found in many insects and later discuss avenues to improve on interpretability and to learn more diverse behaviors.

The controller is implemented as a graph neural network (GNN) [5–7]. Together with an appropriate graph structure it represents a local model where leg control only depends on the leg's own features, as well as features of neighboring legs, i.e. the first-order neighborhood. As shown in Figure 1(*b*), we construct a graph $\mathcal{G} = (V, E)$, where the nodes V correspond to the legs and the edges E correspond to communication channels between edges (colored arrows). This graph reflects the structure of the inter-leg rules found in biological experiments on the stick insect. Both nodes and edges are parameterized by feature vectors. The node features $\mathbf{x}_v \in \mathbb{R}^{24}$ consist of state information of the respective leg, as well as state information of the torso. The edge features $\mathbf{e}_{u,v} \in \mathbb{R}^2$ depend on the edge direction, specifically: rostrally directed edge (blue) $[1, 0]$, caudally directed edge (orange) $[-1, 0]$, contralateral edge (turquoise)



(a) (b)

Figure 1: (a) Coordination rules that have been found in stick insects, acting in the directions of the arrows. Figure adapted from [3]. (b) Our decentralized controller inspired by the coordination rules on the left. Yellow boxes represent nodes of the graph and leg policy (π). Arrows show graph structure utilized by the policies. Arrow colors denote different edge features. The robot body model shown as shaded schema in the background.

$[0,1]$ and contralateral edge (green) $[0,-1]$. Note that neither node, nor edge features contain identifiers that uniquely identify them, thus the learned model has to learn general state and message representations for successful coordination. To ensure that control of each leg only depends on first-degree neighboring legs, the model is a single-layer GNN implemented as

$$\mathrm{msg}_{u\to v}^{t} = \phi\left(\left(\mathbf{x}_v^t - \mathbf{x}_u^t\right) \parallel \mathbf{e}_{u,v}\right) \tag{1}$$

$$\mathbf{a}_v^t = \theta\left(\mathbf{x}_v^t \parallel \sum_{u\in\mathcal{N}_v} \mathrm{msg}_{u\to v}^{t}\right), \tag{2}$$

where ϕ and θ are trainable multilayer perceptrons (MLPs), $\cdot \parallel \cdot$ denotes vector concatenation and $\mathbf{a}_v^t \in \mathbb{R}^3$ denote the actions of leg v at time t. The policy is trained via proximal policy optimization (PPO) [8] in the actor-critic (A2C) flavor, where the ciritc uses the same architecture as the actor, therefore it is also local. The setting is posed as multi-agent reinforcement learning, with every leg implemented as an individual agent.

3. Results & Discussion

Figure 2 shows a preliminary result of our trained policy. The observed behavior resembles a tripod

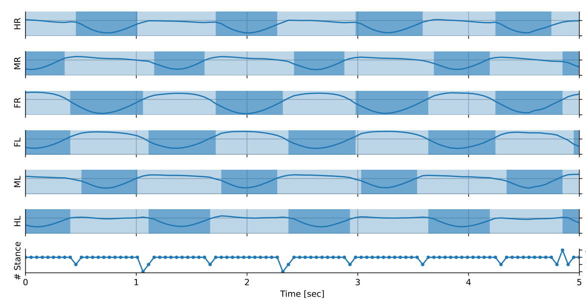


Figure 2: Hip angles (blue curves) of each leg over time shown for 5 seconds (100 simulation steps). Shaded background indicates that leg is in stance (dark) or swing mode (light). Bottom shows the number of legs in stance mode. The shown pattern corresponds to a tripod gait.

gait, where the front-left (FL), middle-right (MR), and hind-left (HL) legs move together while the other legs move in the opposite phase. From our preliminary experiments with different target velocities ($v_{\text{target}} \in [0.1, 0.8]$) we can report that the policy converges consistently to this tripod behavior.

While this simple decentralized architecture replicates biological observations it remains unclear to what extend the rules found in the insect are being implemented. We hypothesize that the messages being sent by the GNN contain much more information than necessary to realize the simple rules discussed above, which might have an adverse effect on learning more diverse walking gaits. Furthermore, rules are only active at very distinct situations, e.g. when the sending leg is currently in swing mode (c.f. rule 1 in [1]). Such mechanics are not explicitly built into our model. Adding an attention mechanism as in graph attention networks [9] to limit information exchange could yield a more interpretable model w.r.t. rule learning and also foster learning.

4. Conclusion

We introduced the idea to learn leg coordination behavior exhibited by insects using graph neural networks. The preliminary results show the possibility to learn a stable tripod gait. In future work, we will investigate 1) how we can make the model more transparent with regards to the coordination rules and 2) how to promote more disverse walking behaviours with such a method.

Acknowledgments

This research was supported by the research training group "Dataninja" (Trustworthy AI for Seamless Problem Solving: Next Generation Intelligence Joins Robust Data Analysis) funded by the German federal state of North Rhine-Westphalia.

References

[1] Holk Cruse. What mechanisms coordinate leg movement in walking arthropods? *Trends in Neurosciences*, 13(1):15–21, January 1990. ISSN 01662236. doi: 10.1016/0166-2236(90)90057-H. URL https://linkinghub.elsevier.com/retrieve/pii/016622369090057H.

[2] Holk Cruse, Thomas Kindermann, Michael Schumm, Jeffrey Dean, and Josef Schmitz. Walknet—a biologically inspired network to control six-legged walking. *Neural Networks*, 11(7):1435–1447, October 1998. ISSN 0893-6080. doi: 10.1016/S0893-6080(98)00067-7. URL https://www.sciencedirect.com/science/article/pii/S0893608098000677.

[3] Malte Schilling, Andrew Melnik, Frank W. Ohl, Helge J. Ritter, and Barbara Hammer. Decentralized control and local information for robust and adaptive decentralized Deep Reinforcement Learning. *Neural Networks*, 144:699–725, December 2021. ISSN 0893-6080. doi: 10.1016/j.neunet.2021.09.017. URL https://www.sciencedirect.com/science/article/pii/S0893608021003671.

[4] Malte Schilling, Kai Konen, Frank W Ohl, and Timo Korthals. Decentralized deep reinforcement learning for a distributed and adaptive locomotion controller of a hexapod robot. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5335–5342. IEEE, 2020.

[5] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605.

[6] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

[7] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry, 2017.

[8] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

[9] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2018.