

Effect of Animat Complexity on the Evolution of Hierarchical Control

Jared M. Moore
School of Computing and Information
Systems
Grand Valley State University
Allendale, Michigan
moorejar@gvsu.edu

Anthony J. Clark
Computer Science Department
Missouri State University
Springfield, Missouri
anthonyclark@missouristate.edu

Philip K. McKinley
Dept of Computer Science and
Engineering
Michigan State University
East Lansing, Michigan
mckinley@cse.msu.edu

ABSTRACT

Animal movements are realized by a combination of high-level control from the nervous system and joint-level movement provided by the musculoskeletal system. The digital muscle model (DMM) emulates the low-level musculoskeletal system and can be combined with a high-level artificial neural network (ANN) controller forming a hybrid control strategy. Previous work has shown that, compared to ANN-only controllers, hybrid ANN/DMM controllers exhibit similar performance with fewer synapses, suggesting that some computation is offloaded to the low-level DMM. An open question is how the complexity of the robot, in terms of the number of joints, affects the evolution of the ANN control structure. We explore this question by evolving both hybrid controllers and ANN-only controllers for worm-like animats of varying complexity. Specifically, the number of joints in the worms ranges from 1 to 12. Consistent with an earlier study, the results demonstrate that, in most cases, hybrid ANN/DMM controllers exhibit equal or better performance than ANN-only controllers. In addition, above a threshold for animat complexity (number of joints), the ANNs for one variant of the hybrid controllers have significantly fewer connections than the ANN-only controllers.

CCS CONCEPTS

•Computing methodologies → Evolutionary robotics;

KEYWORDS

Evolutionary robotics, artificial neural networks, digital muscle model, animats, controller evolution

1 INTRODUCTION

Animals perform complex movements requiring coordination between brain and body. Control signals from the brain propagate through the nervous system to muscles, producing movement. In addition to control signals, the bones, tendons, and muscles that surround a joint influence motion [1, 2, 7, 25]. For example, the physical configuration of tendons in the human hand produces a type

of “anatomical computation” moving fingers without precise input from the neural system [28].

In robotic systems, joint control is typically handled quite differently. Individual joint actuators receive commands directly from a high-level controller. Yet most robotic systems do not possess the agility and dexterity of natural organisms.

In this paper, we evaluate a bio-inspired hierarchical approach to robot control [16]. Unlike typical robotic controllers, control is realized by the interaction between high-level artificial neural network (ANN) and a low-level construct called the *digital muscle model* (DMM) [15]. In this model, a joint is controlled by multiple muscle nodes that define a collective response to an input signal. Digital muscles provide an evolvable mapping between an input signal and joint response. This hybrid control strategy might reduce demand on the high-level controller, as suggested by previous studies comparing ANN-only controllers to ANN/DMM controllers and examining the resulting ANNs [16].

We previously [16] proposed the hierarchical ANN/DMM approach to control and showed that it is effective in different types of systems, including legged animats. We also observed that the evolved ANNs in hierarchical controllers are often simpler (had fewer connections) than those in evolved ANN-only controllers for the same systems, implying a possible offloading of control from the ANN to the DMM. However, our investigations did not examine whether the number of joints in the animat affects the evolution of this characteristic.

We address this question by exploring how the number of joints in an animat influences complexity of the high-level controller. As shown in Figure 1, the animat consists of a fixed length “worm,” subdivided into equally sized segments based on the number of joints in a given configuration. Controllers consist of a high-level ANN paired with a low-level DMM. Three separate control combinations are tested, one an ANN-only controller and two hybrid ANN/DMM controllers with different connectivity between the two control layers. The controllers are evolved for worm-like animats, where the number of joints ranges from 1 to 12.

The contributions of this work are as follows. First, the results support earlier studies that evolved hybrid ANN/DMM controllers have the same or better, performance compared to their ANN-only counterparts. Second, ANNs in hybrid controllers exhibit fewer connections than those in ANN-only controllers, but only for animats whose number of joints is above a threshold, potentially leading to more efficient controllers. Finally, we found that in contrast to the number of connections, the number of hidden nodes in the hybrid controllers is consistently higher than ANN-only controllers.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

GECCO '17, Berlin, Germany

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-1-4503-4920-8/17/07...\$15.00

DOI: <http://dx.doi.org/10.1145/3071178.3071246>

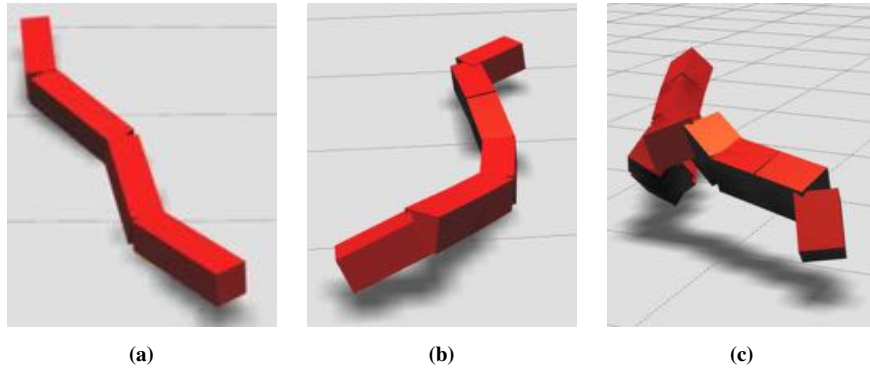


Figure 1: Three worm-like robots. The overall shape and mass of the robot remains constant throughout the different configurations. (a) Three-joint, (b) five-joint, and (c) ten-joint robot.

2 BACKGROUND AND RELATED WORK

Evolutionary robotics [6, 17, 23] applies concepts from natural evolution to the design of robotic systems. Evolved controllers have been applied to problems such as rocket guidance [10], quadrupedal gaits [4], and robot design for space exploration [21]. Morphology also has a strong influence on overall performance and even performs a type of “morphological computation” [19] contributing to the behavior of a system. For example, Rieffel et al. [20] demonstrated that by evolving the morphology of a tensegrity robot it can exhibit effective locomotion without a high-level controller. Control and morphology can be evolved simultaneously [30] to produce highly integrated optimized systems [11, 12].

Introducing modularity as a factor in either the ANN controller, or morphology, can increase performance of robotic systems. Valsalam et al. [29] demonstrated that enforcing ANN controller modularity (encouraging symmetry and gait robustness) produces effective gaits in quadrupedal animats. Furthermore, modularity has been shown to play an important role in increasing performance in game playing agents [22], robot controllers [5, 18], and robust animats [27]. Additionally, the body plays an important role in the evolution of controllers. For instance, Bongard et al. [3] demonstrated that morphology encourages the emergence of neural modularity. In this work, we employ a computational model of multi-level control that emulates modularization of control between aspects of the brain (ANN) and body (DMM).

Muscle-inspired control models have produced effective bipedal gaits in humanoid [31] and other simulated animats [8]. Lessin et al. [13] applied an evolutionary approach to the development of complex behaviors in simulated animats based on a custom muscle-inspired actuation model. Further research has shown that some control decisions can be offloaded to the musculoskeletal configuration of the animat itself [14]. These models, however, directly emulate the behavior and physical characteristics of muscles in simulation. In contrast, the DMM is an abstract model borrowing concepts from natural muscles (spatial position and activation functions) yet is suitable for conventional robotic actuators (servo motors).

3 METHODS

Digital Muscle Model. The DMM, illustrated in Figure 2, is a low-level control model that maps an input control signal to an output activation. The central component of the DMM is the muscle

group, consisting of digital muscle nodes, governing the behavior of an individual joint. Figure 2 shows an individual muscle group, as applied to a joint between two segments, consisting of four muscle nodes depicting how the response of the muscle nodes influence the behavior of the joint. Each muscle node has an evolvable position relative to the joint and activation function. For this work, each joint group has four muscle nodes, but it can be changed in the model.

Figure 3 shows the process of converting sensory input to actuator output for hybrid ANN/DMM controllers. First, the sensory information is fed into the high-level ANN controller, which then passes a signal to each joint group. Depending on the hybrid controller, ANN outputs are either connected to the joint group, or individual muscle nodes. For joint groups with one ANN output, the same control signal is passed to each muscle node. The internal behavior of a muscle node maps the signal to a joint actuation. A group of muscle nodes define the direction and angle of movement for a single joint. Full details of the DMM are available in [15].

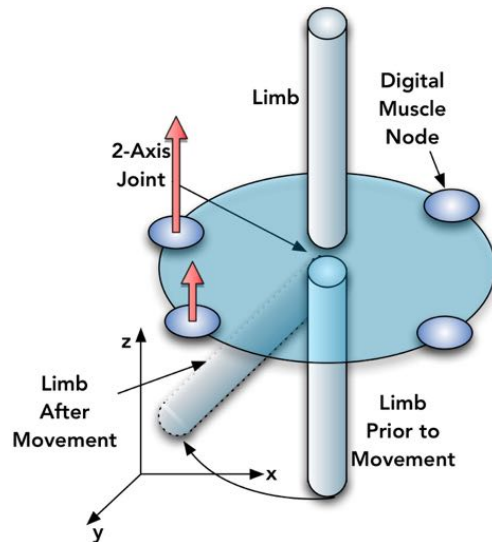


Figure 2: A single DMM digital muscle group controls one joint in an animat. Digital muscle nodes around a joint (their positions are evolved) exert a pulling force based on the input signal from a high-level controller and their internal activation function. (Adapted from [15].)

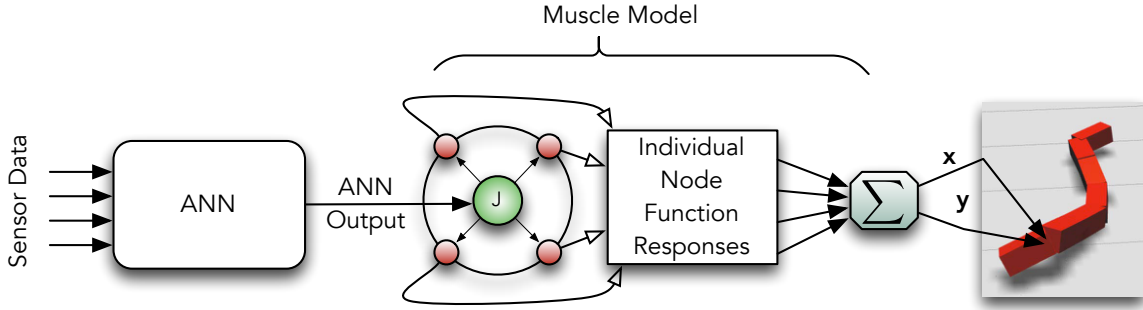


Figure 3: An ANN/DMM controller diagram showing how inputs to the ANN flow through the high-level controller into one muscle group controlling one joint in the animat. (Adapted from [16].)

Hybrid ANN/DMM Control. The ANN controller component is evolved with the NEAT Algorithm [26] using parameters presented in Table 1. Pairing DMM controllers with a high-level ANN is shown in Figure 3. The ANN provides closed-loop control by adjusting outputs based on sensor data (e.g. touch and joint-angle sensors). In this study, each muscle group contains four muscle nodes. Together, the ANN and DMM provide high and low-level control, respectively.

Robot Platform. Throughout all experiments conducted in this paper, the overall length, width, height, and mass of the animat remain fixed. As the number of joints increases, the body is divided into increasingly shorter segments. Each joint is a 2-degree of freedom (DOF) hinge allowing for movements perpendicular to the longitudinal axis of the worm with a range of motion of $\pm 90^\circ$ in each axis. The segments, simulated as cuboids with sharp corners, are allowed to intersect with each other. Inputs to the ANN comprise two angle sensors per joint, a touch sensor for each body segment, and a bias input typical for most ANNs. A touch sensor triggers when any part of the associated body segment contacts the ground, irrespective of orientation.

Simulation Environment. The Open Dynamics Engine (ODE) [24], a 3D rigid body physics simulation engine, is used to evaluate behaviors. ODE simulates friction, gravity, and collisions between bodies. Joint actuation is achieved by specifying the desired angular velocity of each hinge. An individual evaluation is conducted for 10 seconds of simulation time with a timestep of 0.005s. The environment is a flat, high-friction surface minimizing slippage between the animat and substrate.

Evolutionary Setup. Three separate treatments are conducted with a setup similar to those found in [16]: singly-connected ANN/DMM (SC), individually-connected ANN/DMM (IC), and ANN-only controller (ANN). The ANN treatment employs an ANN to directly command the animat’s joints. The SC treatment maps a single output from the ANN to each joint, with all four muscle nodes in the group receiving the same signal (depicted in Figure 2). In contrast, the IC treatment has a unique ANN output for each muscle node in an animat (not shown).

Populations comprise 120 individuals and are evolved for 1000 generations. We conduct 20 replicate runs per treatment, each with a unique starting seed for randomization. An individual’s fitness is defined as the Euclidean distance from the origin to the average (X,Y)

position of the segments after 10 seconds of simulation time. Each individual’s genome includes information about the high-level ANN and low-level DMM. NEAT is used to evolve the ANN component of an individual genome while the DMM component is subjected to mutation and crossover operators that modify the activation function and spatial position of the muscle nodes.

Parameter	Value	Parameter	Value
Compatibility Threshold	5.0	Mutation Rate	0.33
Young Age Threshold	15	Mut Weights Prob	0.90
Species Stagnation	1000	Weight Mut Rate	0.75
Old Age Threshold	35	Max Weight	20
Min Species	1	Add Neuron Prob	0.4
Max Species	25	Add Link Prob	0.4
Recurrent Prob	0.25	Rem Link Prob	0.05
Crossover Rate	0.75		

Table 1: NEAT Algorithm Parameters

4 EXPERIMENTS AND RESULTS

4.1 Evolved Gaits

A variety of unique gaits evolve across the different configurations and replicates. Although the focus of this investigation is not on the characteristics of individual gaits, we briefly review the types of locomotion observed for this worm-like animat. Figure 4 shows a sample of three gaits, one from each treatment. Videos of the gaits are available at <https://youtu.be/CgSG93-D8eo>. We note that all three controller treatments evolve effective gaits for each of the twelve configurations. Many different behaviors evolve including, among others, folding (middle of the worm hinges while ends act as feet), hopping (one end curls and acts as a primitive leg), and rolling (robot curls into a wheel).

4.2 Analysis

Figure 5 plots the distribution of fitnesses for the farthest traveling individuals, one per replicate, across the three treatments. Here we find that the highest performers arise out of the hybrid ANN/DMM controllers in low (≤ 5) and high (≥ 8) joint robots. For animats with 6 and 7 joints, ANN-only controllers evolve the farthest traveling individuals. Table 2, provides all pairwise comparisons using a Wilcoxon Rank Sum Test between treatments. A prior investigation [15] showed that evolved DMM controllers driven by a sinusoid

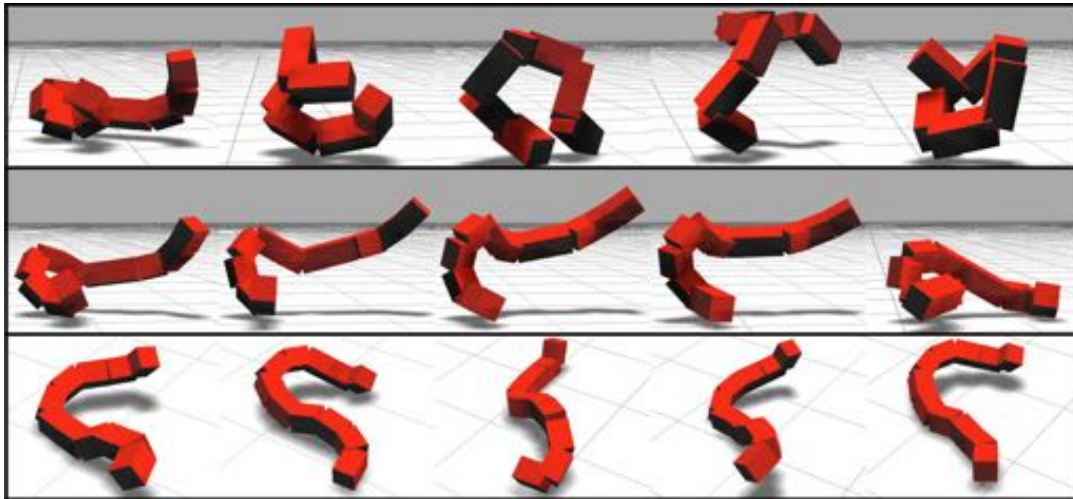


Figure 4: A sample of three gaits, one from each treatment. (Top) An ANN-only evolved controller that exhibits a “rolling” gait, curling and unfolding to produce movement. (Middle) A singly-connected controller with a “hopping” gait; the rear of the worm acts as a primitive leg. (Bottom) An individually-connected controller with a “walking” gait; the ends of the robot act as primitive legs.

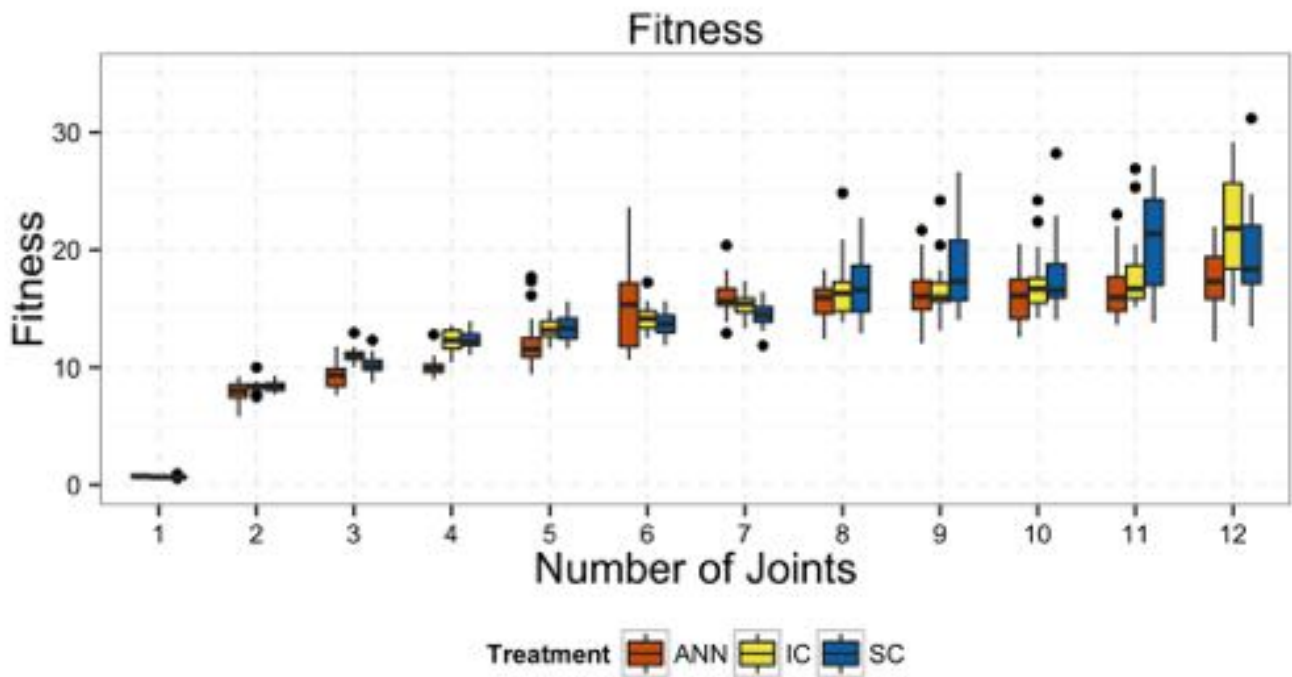


Figure 5: Boxplot showing the fitness of the farthest traveling individual per replicate for the three treatments across the different number of joints. The hybrid ANN/DMM controllers tend to have higher fitnesses than the best ANN controllers.

controller facilitated quadrupedal locomotion. Here, in robots with 8 joints and greater, we hypothesize that the hybrid ANN/DMM controllers are able to establish basic movements through the DMM allowing the ANN component to provide control signals for these low-level movements. Whereas, ANN-only controllers instead need to evolve a control strategy for each joint, potentially making the problem more difficult, especially as the number of joints increases.

This difference in control structure between hybrid and ANN-only control could explain the performance differences observed in higher joint animats.

Figure 6 plots the number of connections versus fitness, grouped by the number of joints in the robot, for the farthest traveling individual per replicate in each of the three treatments.

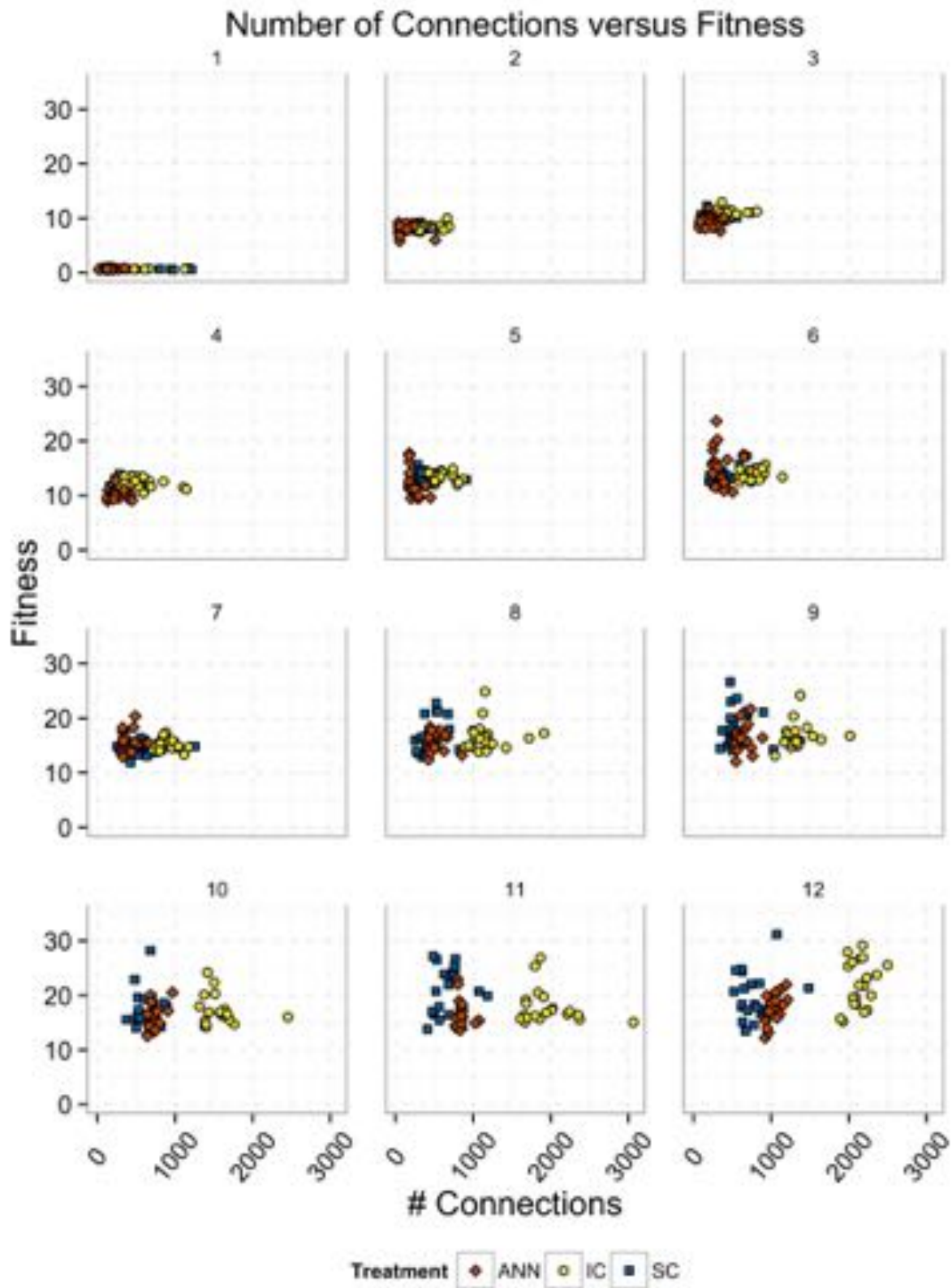


Figure 6: Number of connections versus fitness in the farthest traveling individuals from each replicate run across the twelve joints.

The number of connections in the evolved ANNs for the farthest traveling individuals per replicate varies considerably across the three treatments. In general, IC controllers have the highest growth in number of connections as the number of joints increases. This is not unexpected, considering the high number of ANN outputs due to connecting to each individual muscle node in the animat. SC and ANN-only controllers both exhibit increases in network connections, but at reduced rates. By 11 and 12 joints, the SC controllers have the fewest number of connections among the three treatments, yet include high performing individuals.

Figure 7 plots distributions for the number of connections in the farthest traveling individuals for each of the number of joints examined in this study. For 1 to 7 joints, ANN-only controllers have the lowest number of connections. However, for 9 joints and higher, the SC controllers have significantly fewer connections in their evolved networks (P-values shown in Table 2). As shown in Figure 5, above 8 joints the SC controllers achieve similar or higher fitnesses than the ANN-only controllers, while having fewer connections. This result suggests that the ANN “offloads” some control functionality to the DMM while maintaining similar performance. Furthermore, this result indicates a point where an ANN/DMM controller is effective for locomotion (8 joints and higher). These results are similar to those observed in the quadruped and hexapod platforms examined in [16].

Figure 8 plots the number of hidden nodes across the range of joints. All three controllers exhibit a relatively constant number of hidden nodes. This result contrasts with that for the number of connections, which steadily increases with the number of joints. ANN-only controllers have the lowest number of hidden nodes across all joints. We speculate that this result is related to the fact that hidden nodes typically act as computational units, while connections facilitate information transfer in ANNs [32].

These results suggest that there is a certain threshold of hidden nodes required in any evolved ANN, regardless of the number of joints. In the case of the hybrid controllers, the hidden nodes may provide alternative computations compensating for the limited communication capacity between the ANN and the low-level DMM control. Furthermore, the decreased number of connections in combination with the hidden nodes may indicate that independent computations within the ANN are more prevalent in hybrid controllers when compared to ANN-only controllers. A neuron performs computation on the inputs, producing an output, while connections transfer outputs of the neurons throughout an ANN [32].

4.3 Singly- versus Individually-Connected

Consistent with an earlier study [16], the SC controllers offer similar performance to the IC strategy, while requiring fewer ANN outputs and, therefore, less connectivity between ANN and DMM. From a computational perspective, smaller ANNs will require fewer resources to calculate command outputs. In the case of the worm, the fitnesses between SC and IC controllers are significantly different only for 3, 7, and 11 joints ($p < 0.001$, $p = 0.0143$, $p = 0.01217$). However, the number of connections in these evolved networks is significantly different for all robots except those with 1 joint ($p = 0.9042$). As shown in Figure 7, the number of connections in the SC controllers grows at a slower rate than that of the IC controllers. A

slower growth rate, as observed in the SC controllers, could allow for an animat with a higher number of joints than would be possible with IC controllers. High ANN complexity will reduce computational efficiency.

	Comp:	1	2	3	4	5	6
Fitness:	SC v IC	0.265	0.862	0.001	1	0.862	0.547
	SC v ANN-only	0.030	0.043	0.002	0.001	0.004	0.314
	ANN-only v IC	0.121	0.072	0.001	0.001	0.002	0.495
Num Con:	SC v IC	0.904	0.028	0.003	0.001	0.001	0.001
	SC v ANN-only	0.006	0.001	0.020	0.010	0.001	0.402
	ANN-only v IC	0.001	0.001	0.001	0.001	0.001	0.001
Num Hid:	SC v IC	0.465	0.449	0.083	0.695	0.978	0.310
	SC v ANN-only	0.002	0.001	0.001	0.001	0.001	0.001
	ANN-only v IC	0.001	0.001	0.001	0.001	0.001	0.001
	Comp:	7	8	9	10	11	12
Fitness:	SC v IC	0.014	0.602	0.232	0.698	0.012	0.091
	SC v ANN-only	0.001	0.165	0.127	0.091	0.001	0.076
	ANN-only v IC	0.211	0.429	0.621	0.211	0.068	0.002
Num Con:	SC v IC	0.001	0.001	0.001	0.001	0.001	0.001
	SC v ANN-only	0.001	0.140	0.007	0.003	0.001	0.001
	ANN-only v IC	0.001	0.001	0.001	0.001	0.001	0.001
Num Hid:	SC v IC	0.005	0.137	0.850	0.756	0.120	0.473
	SC v ANN-only	0.001	0.001	0.001	0.001	0.001	0.001
	ANN-only v IC	0.001	0.001	0.001	0.001	0.001	0.001

Table 2: P-values of pairwise comparison using a Wilcoxon Rank Sum Test for the farthest traveling individual per replicate from the three treatments. These numbers signify whether there is a significant difference between the two treatments. Reference Figures 5, 7, and 8. The three metrics are listed on the left: fitness, number of connections and number of hidden nodes in the evolved networks. Treatments are abbreviated as follows: (SC) singly-connected, (IC) individually-connected, and ANN-only.

5 CONCLUSIONS

In this paper, we investigated a hierarchical animat control strategy focusing on how the number of joints relates to ANN complexity and performance among different configurations. As with earlier studies of quadrupedal and hexapedal locomotion, hybrid ANN/DMM controllers exhibit similar performance to ANN-only controllers for the majority of test cases. However, as the number of joints in a robot increases, the farthest traveling hybrid controllers outperform their ANN-only counterparts, while exhibiting fewer connections in evolved ANNs. This result suggests that the ANN is offloading some control functionality to the DMM, similar to theories of biological control [9].

Furthermore, a single connection (SC) between the ANN and each muscle group is sufficient to produce effective locomotion. This configuration also leads to fewer connections within the ANN while maintaining performance similar to that of the individually connected (IC) configuration. Such modularization in control might free the high-level controller to focus on tasks other than governing low-level movement of joints. Future studies will investigate how/if this result generalizes to other platforms and behaviors. Considering these, and previous [16] results, we expect this finding to generalize to other platforms as it has been shown in quadrupedal, hexapedal, and now worm-like animats.

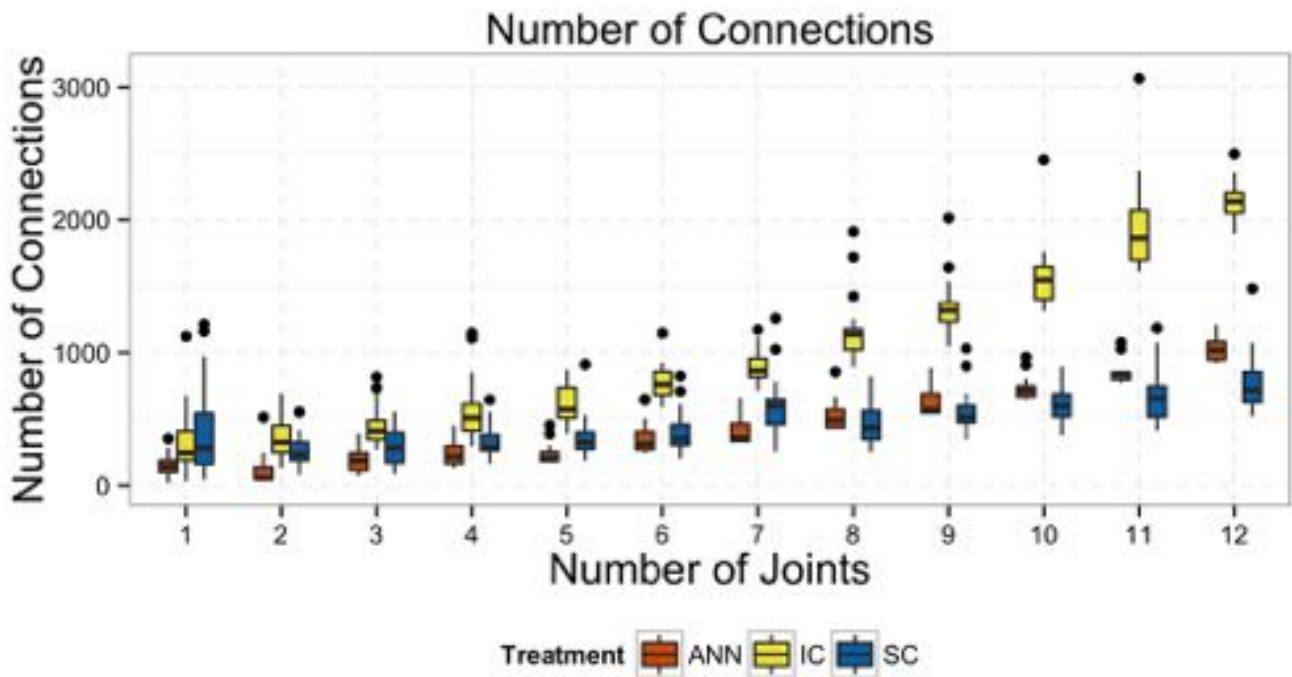


Figure 7: Number of connections for the farthest traveling individuals from 20 replicate runs per each number of joints across the three treatments. Differences are statistically significant for all except singly- versus individually-connected one joint ($p = 0.9042$) and singly-connected versus ANN-only six ($p = 0.4017$) and eight joints ($p = 0.1404$).

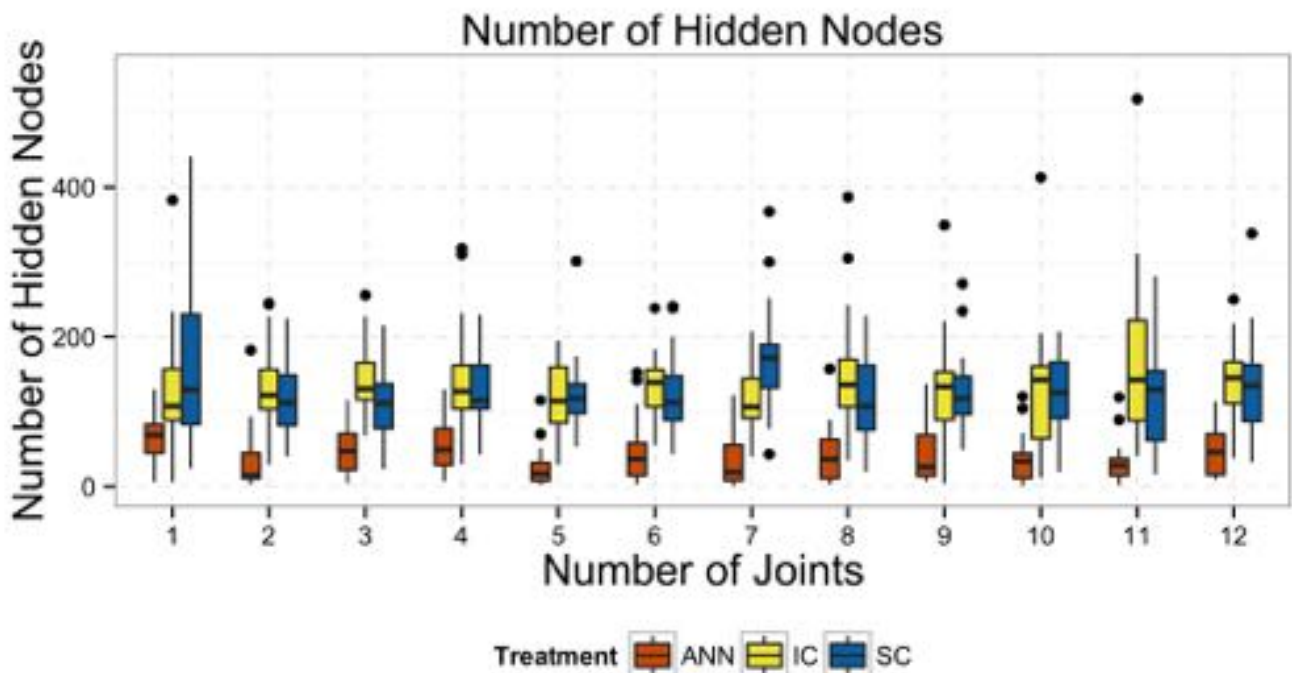


Figure 8: Number of hidden nodes for the farthest traveling individuals from 20 replicate runs per each number of joints across the three treatments. Differences are statistically significant for all ANN/DMM versus ANN-only controllers. There are no significant differences in the number of hidden nodes for singly- and individually-connected controllers except for 7 joints ($p = 0.0047$).

ACKNOWLEDGMENTS

The authors gratefully acknowledge the contributions and feedback provided by Xiaobo Tan, Craig McGowan, and members of the BEACON Center at Michigan State University. This work was supported in part by National Science Foundation grants CNS-1059373, CNS-0915855, and DBI-0939454, and by a grant from Michigan State University.

REFERENCES

- [1] R. Alexander and Alexandra Vernon. 1975. The mechanics of hopping by kangaroos (Macropodidae). *Journal of Zoology* 177, 2 (1975), 265–303.
- [2] Kellar Autumn, Metin Sitti, Yiching A. Liang, Anne M. Peattie, Wendy R. Hansen, Simon Sponberg, Thomas W. Kenny, Ronald Fearing, Jacob N. Israelachvili, and Robert J. Full. 2002. Evidence for van der Waals adhesion in gecko setae. *Proceedings of the National Academy of Sciences* 99, 19 (2002), 12252–12256.
- [3] Josh C. Bongard, Anton Bernatskiy, Ken Livingston, Nicholas Livingston, John Long, and Marc Smith. 2015. Evolving Robot Morphology Facilitates the Evolution of Neural Modularity and Evolvability. In *Proceedings of the 2015 Genetic and Evolutionary Computation Conference*. ACM, Madrid, Spain, 129–136.
- [4] Jeff Clune, Benjamin E. Beckmann, Charles Ofria, and Robert T. Pennock. 2009. Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computation*. Trondheim, Norway, 2764–2771.
- [5] S. Doncieux and J.-A. Meyer. 2004. Evolving Modular Neural Networks to Solve Challenging Control Problems. In *Proceedings of the Fourth International ICSC Symposium on Engineering of Intelligent Systems (EIS 2004)*. Madeira, Portugal, 1–7.
- [6] Dario Floreano, Phil Husbands, and Stefano Nolfi. 2008. Evolutionary Robotics. In *Handbook of Robotics*. Springer Verlag, Berlin.
- [7] Rudolf M. Fuchsli, Andrej Dzyakanchuk, Dandolo Flumini, Helmut Hauser, Kenneth J. Hunt, Rolf H. Luchsinger, Benedikt Reller, Stephan Scheidegger, and Richard Walker. 2013. Morphological Computation and Morphological Control: Steps Toward a Formal Theory and Applications. *Artificial Life* 19, 1 (2013), 9–34.
- [8] Thomas Geijtenbeek, Michiel van de Panne, and A. Frank van der Stappen. 2013. Flexible Muscle-based Locomotion for Bipedal Creatures. *ACM Transactions on Graphics* 32, 6 (2013), 1–11.
- [9] Simon F. Giszter, A. Mussa-Ivaldi, and Emilio Bizzi. 1993. Convergent force fields organized in the frog’s spinal cord. *Journal of Neuroscience* 13, 2 (1993), 467–491.
- [10] Faustino Gomez and Risto Miikkulainen. 2003. Active Guidance for a Finless Rocket Using Neuroevolution. In *Proceedings of the 2003 Genetic and Evolutionary Computation Conference*. Chicago, Illinois, USA, 2084–2095.
- [11] J.D. Hiller and H. Lipson. 2010. Evolving Amorphous Robots. In *Proceedings of the Twelfth International Conference on Artificial Life*. Odense, Denmark, 717–724.
- [12] Gregory S. Hornby and Jordan B. Pollack. 2001. Body-Brain Co-evolution Using L-systems as a Generative Encoding. In *Proceedings of the 2001 ACM Genetic and Evolutionary Computation Conference*. Morgan Kaufmann, San Francisco, California, USA, 868–875.
- [13] Dan Lessin, Don Fussell, and Risto Miikkulainen. 2013. Open-Ended Behavioral Complexity for Evolved Virtual Creatures. In *Proceedings of the 2013 ACM Genetic and Evolutionary Computing Conference*. ACM, Amsterdam, Netherlands, 335–342.
- [14] Dan Lessin, Don Fussell, and Risto Miikkulainen. 2014. Trading Control Intelligence for Physical Intelligence: Muscle Drives in Evolved Virtual Creatures. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*. ACM, Vancouver, BC, Canada, 705–712.
- [15] Jared M. Moore and Philip K. McKinley. 2014. Evolving Joint-Level Control with Digital Muscles. In *Proceedings of the 2014 ACM Genetic and Evolutionary Computing Conference*. ACM, Vancouver, BC, Canada, 209–216.
- [16] Jared M. Moore and Philip K. McKinley. 2014. Investigating Modular Coupling of Morphology and Control with Digital Muscles. In *Proceedings of the 14th International Conference on the Simulation and Synthesis of Living Systems*. ACM, New York, NY, USA, 148–155.
- [17] Stefano Nolfi and Dario Floreano. 2000. *Evolutionary Robotics: The Biology, Intelligence and Technology of Self-Organizing Machines*. The MIT Press.
- [18] Frank Pasemann, Uli Steinmetz, Martin Hulse, and Bruno Lara. 2001. Robot control and the evolution of modular neurodynamics. *Theory in Biosciences* 120, 3-4 (2001), 311–326.
- [19] Chandana Paul. 2006. Morphological computation: A basis for the analysis of morphology and control requirements. *Robotics and Autonomous Systems* 54, 8 (2006), 619 – 630.
- [20] John A. Rieffel, Francisco J. Valero-Cuevas, and Hod Lipson. 2010. Morphological communication: Exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of The Royal Society Interface* 7, 45 (April 2010), 613–621.
- [21] M. Rommerman, D. Kuhn, and F. Kirchner. 2009. Robot design for space missions using evolutionary computation. In *IEEE Congress on Evolutionary Computation*. Trondheim, Norway, 2098–2105.
- [22] Jacob Schrum and Risto Miikkulainen. 2014. Evolving Multimodal Behavior with Modular Neural Networks in Ms. Pac-Man. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation (GECCO ’14)*. ACM, Vancouver, BC, Canada, 325–332.
- [23] Karl Sims. 1994. Evolving 3D morphology and behavior by competition. *Artificial Life* 1, 4 (1994), 353–372.
- [24] Russell Smith. 2013. Open Dynamics Engine, <http://www.ode.org/>. (2013). <http://www.ode.org/>
- [25] J C Spagna, D I Goldman, P-C Lin, D E Koditschek, and R J Full. 2007. Distributed mechanical feedback in arthropods and robots simplifies control of rapid running on challenging terrain. *Bioinspiration & Biomimetics* 2, 1 (2007), 9–18.
- [26] Kenneth O. Stanley and Risto Miikkulainen. 2002. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation* 10, 2 (June 2002), 99–127.
- [27] Adam Stanton and Alastair Channon. 2013. Incremental Neuroevolution of Reactive and Deliberative 3D Agents. In *Proceedings of the 13th European Conference on Artificial Life*. York, UK, 341–348.
- [28] F.J. Valero-Cuevas, Jae-Woong Yi, D. Brown, R.V. McNamara, C. Paul, and H. Lipson. 2007. The Tendon Network of the Fingers Performs Anatomical Computation at a Macroscopic Scale. *IEEE Transactions on Biomedical Engineering* 54, 6 (June 2007), 1161–1166.
- [29] Vinod K. Valsalam and Risto Miikkulainen. 2008. Modular neuroevolution for multilegged locomotion. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. ACM, Atlanta, GA, USA, 265–272.
- [30] Barthelemy von Haller, Auke Ijspeert, and Dario Floreano. 2005. Co-evolution of Structures and Controllers for Neobot Underwater Modular Robots. In *Advances in Artificial Life*, Mathieu S. Capcarrere, Alex A. Freitas, Peter J. Bentley, Colin G. Johnson, and Jon Timmis (Eds.). Lecture Notes in Computer Science, Vol. 3630. Springer Berlin Heidelberg, 189–199.
- [31] Jack M. Wang, Samuel R. Hamner, Scott L. Delp, and Vladlen Koltun. 2012. Optimizing Locomotion Controllers Using Biologically-based Actuators and Objectives. *ACM Trans. Graph.* 31, 4, Article 25 (July 2012), 25:1–25:11 pages.
- [32] X. Yao. 1999. Evolving artificial neural networks. *Proc. IEEE* 87, 9 (1999), 1423–1447.