The Evolution of Controller-Free Molecular Motors from Spatial Constraints

Jose David Fernández
Grupo de Estudios en Biomimética,
Departamento de Lenguajes y
Ciencias de la Computación,
Universidad de Málaga,
Málaga, Spain
josedavid@geb.uma.es

René Doursat
Grupo de Estudios en Biomimética,
Departamento de Lenguajes y
Ciencias de la Computación,
Universidad de Málaga,
Málaga, Spain
doursat@geb.uma.es

Francisco J. Vico
Grupo de Estudios en Biomimética,
Departamento de Lenguajes y
Ciencias de la Computación,
Universidad de Málaga,
Málaga, Spain
fjv@geb.uma.es

***Abstract*—Locomotion of robotic and virtual agents is a challenging task requiring the control of several degrees of freedom as well as the coordination of multiple subsystems. Traditionally, it is engineered by top-down design and fine-tuning of the agent's morphology and controller. A relatively recent trend in fields such as evolutionary robotics, computer animation and artificial life has been the *coevolution* and mutual adaptation of the morphology and controller in computational agent models. However, the controller is generally modeled as a complex system, often a neural or gene regulatory network. In the present study, inspired by molecular biology and based on normal modal analysis, we formulate a behavior-finding framework for the design of bipedal agents that are able to walk along a filament and have *no explicit control system*. Instead, agents interact with their environment in a purely reactive way. A simple mutation operator, based on physical relaxation, is used to drive the evolutionary search. Results show that gait patterns can be evolutionarily engineered from the spatial interaction between precisely tuned morphologies and the environment.**

***Index Terms*—morphological computation, elastic network model, behavior-finding**

I. Introduction

Engineers have made remarkable progress in their ability to design complex products. However, current engineering practice still favors a top-down approach, where the main problem is manually divided into smaller ones in order to maintain the overall complexity reasonably manageable. This procedure is rather loosely defined and ultimately relies on human expertise and creativity, which are skills that typically involve high costs, are unreliable and are difficult to formalize. Moreover, the ever increasing complexity of current engineered systems and robustness requirements is reaching the feasibility limits of the current paradigm, forcing the implementation of new engineering methodologies.

Inspired by the biological evolution and morphogenesis of organisms, recent advances in the discipline of evolutionary computation propose a radically different approach. Genetic algorithms combined with artificial development mechanisms operate over a population of individuals encoded by *genomes* that govern a morphogenetic process producing self-constructed designs [1]. That is, the genome is not a blueprint of the design, but the set of instructions that indirectly build it. The evolutionary operations (mutations and crossover) are applied to the developmental generative process that build the design, not to the design itself. This approach has been shown to overcome the issues of scalability, adaptability, and evolvability present in traditional evolutionary algorithms (based on a genomic representation that encodes the design in a explicit way) when applied to complex problems [2]. As a result, evolutionary developmental algorithms have been tried in a wide range of design problems, including the structure and controller of robots [3], digital creatures in Artificial Life studies [4, 5, 6, 7, 8], and computer animated characters [9]. In almost all models, however, the control system is fairly complex (often based on some kind of recurrent neural network), and in many cases, we believe, unnecessarily so.

In a seminal work [10], Chandana Paul demonstrated that a whole body-control system is able to perform more complex computations than the control system alone. This observation spawned the concept of *morphological computation*—a design methodology for robotic-like agents to exploit the dynamics of interaction between the body and the control system, resulting in minimal control systems. Several applications have been proposed in the field of robotics, including the design of semipassive bipedal robots [11], tensegrity robots whose complex, coupled non-linear dynamics are harnessed to generate a gait pattern with minimal control [10], path-following agents [12], and robots with open-loop control systems and minimal numbers of degrees of freedom that can self-stabilize into fast gait patterns and generate diverse behaviors through the interaction between body and control system [13].

We present here a framework to generate bipedal agents that can walk along a filament, taking inspiration from biological molecular motors. Toward this goal, we apply a simple evolutionary heuristic based on normal modal analysis [14] and a behavior-finding strategy [12]. Our work can be construed as "morphological computation" in two ways. First, the behavior of the agents is not driven by a complex, network-based control system, but emerges from their spatial characteristics. Second, we use a simple and explicit genetic representation, combined with a physics-based mutation operator able to

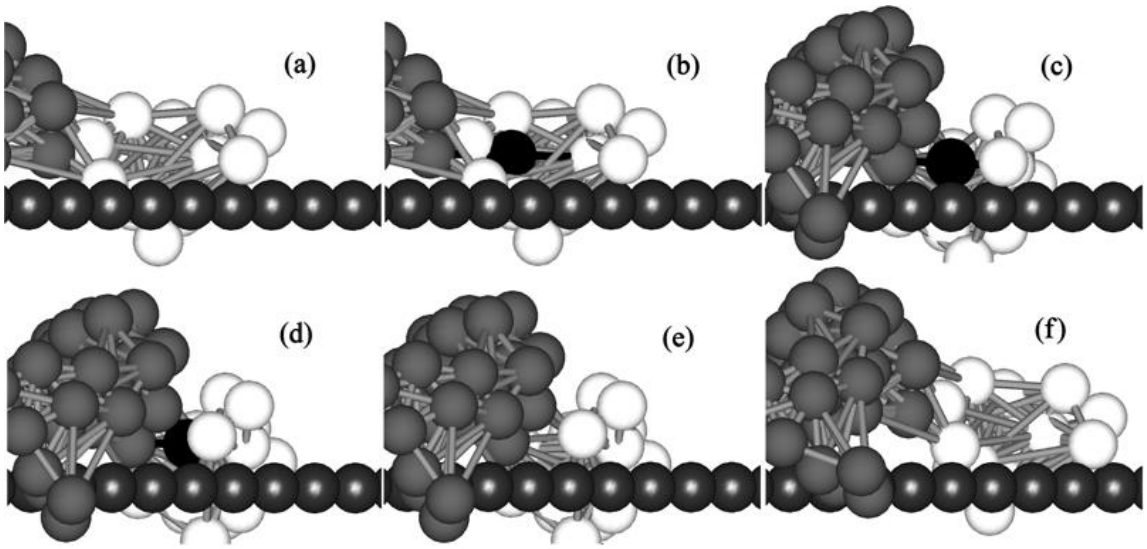

Figure 1. Working cycle of a motor template.

induce *coordinated changes* in the agents' structure. In this way, we take full advantage of the spatial and geometric nature of the genotype.

II. Methods

The model is motivated by biological molecular motors, such as the enzymes myosin, kinesin and dynein, capable of transforming chemical energy into mechanical work. Breaking down *ATP* molecules for power, these molecular motors can effectively *walk* along cellular filaments [15]. They are composed of one or two *motor heads*, each comprising a *catalytic core* (the site where ATP molecules attach) and a *docking site* (the site where the motor attaches to the filament). Each motor head undergoes a cycle (working cycle) of shape changes (*conformational changes*), powered by the energy stored in ATP molecules. The docking site cyclically attaches and detaches from the filament in a coordinated way, allowing the motor to advance through the filament.

Molecular motors can be construed as nanoscale robotic agents. The control system is implicitly defined in the specific biochemical interactions between the molecular motor, the ATP molecules, and the filament; in this way, their morphologies canalize the movements and the function of the motors [16]. Indeed, molecular motors represent a clear example of morphological computation. Taking inspiration from this observation, we have built a framework based on evolutionary optimization to design robotic agents that function in a way similar to molecular motors. We call these agents *motor templates*, following our earlier work on this topic [17]. A motor template represents the structure of a plausible protein. It is modeled by a folded chain of vertices, in which elastic links are established between two vertices if and only if their distance is less than a given threshold [18]. Thus the whole object constitutes a 3D mass-spring network. While modeling molecular motors with mass-spring networks may seem simplistic, it can be justified theoretically: for most proteins, including many molecular motors, the dynamics is mainly dictated by their overall structure rather than their specific biochemical compositions [19].

A. Motor templates

A template has two motor heads, each one endowed with a catalytic core and a docking site. The catalytic core is defined as a set of two nodes in the network. When an ATP molecule binds to the core, it is placed exactly in the middle of the two vertices, connected by a spring to each vertex in the pair. These springs are stretched to model the change in potential energy brought by the ATP molecule (this mechanism has been used in other studies, as [18]). The docking site is modeled as a set of nodes that can attach and detach from the filament. The working cycle of a motor head can be described as a reactive finite-state machine with four states:

1) *Sticky* state: The docking site is not in contact with the filament, and the catalytic core is empty (Figure 1a). This state ends when any node of the docking site touches the filament: the node becomes fixed to the filament, and an ATP molecule is bound to the catalytic core with stretched springs (Figure 1b). Then, the motor head transitions to the next state.
2) *Bound* state: The stretched springs introduced in the transition to this state induce a conformational change (Figure 1c), while the docking site remains firmly attached to the filament, resulting in a conformational change. After a fixed amount of time passes, the motor head transitions to the next state.
3) *Nonsticky* state: the nodes of the docking site detach from the filament, but remain in contact with it. If the activity of the other motor head or residual elastic forces drive the docking site out of contact with the filament (Figure 1d), the ATP is expulsed from the catalytic core, deleting the associated springs (Figure 1e). Then, the motor head transitions to the next state.

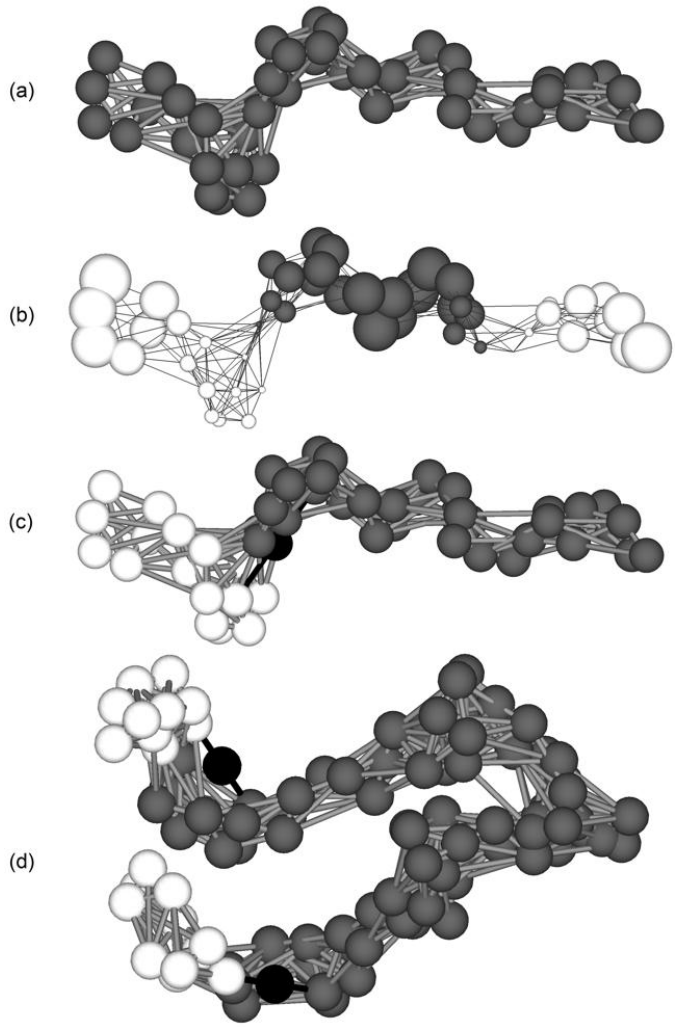

Figure 2. A mass-spring network (a) is processed to determine its catalytic cores and docking sites. The normal mode associated to the third eigenvector of its Kirchhoff matrix is shown (b). Each vertex is associated to a component of the eigenvector, whose magnitude (size) and sign (white positive, gray negative) conveys information about the vibration of the vertex in that normal mode, splitting the structure into three clusters. A motor head (c) is then composed of a catalytic core (ATP and connecting springs shown in black) placed between a distal cluster and the central one, and a docking site (in white). Finally, a motor template (d) is defined by joining two structures that are mirror images of each other.

4) *Relaxing* state: When the catalytic core becomes empty, the absence of the associated springs triggers another conformational change. After a fixed amount of time passes, the vertices of the docking site regain the ability to get fixed to the filament, and the motor head transitions to the initial state (Figure 1f), completing the cycle. A motor head has *completed* a working cycle when it has passed through all states and is back to the initial one: 1-2-3-4-1.

Simple and elegant theoretical tools that consider proteins as mass-spring networks, such as the Gaussian Network Model (GNM), use normal mode analysis to predict their structural and dynamical properties, and can do so to a surprising extent, including their unfolding pathways [20], their domain decomposition [21], and, in particular, their conformational changes and the position of their catalytic cores [22]. We use a heuristic based in GNM to determine the placement of the docking sites and catalytic cores, which are indirectly encoded in the morphology of the structure. Specifically, to define a motor head (with a catalytic core and a docking site) in the mass-spring network of a template (Figure 2a), we segment the network using the normal mode associated to the third eigenvector of its Kirchhoff matrix [22]. This eigenvector assigns a vibrational amplitude to each node in the network, which can be either positive or negative. In Figure 2b, each node's size and color represent the amplitude and sign, respectively (white is positive, gray is negative). Grouping neighboring nodes with same-sign vibrational amplitudes, three clusters can be defined in most mass-spring networks. There are two interfaces (hinges) between the clusters, such that two of the clusters are distal while the other one is central. As the third eigenvector is associated to a low-frequency normal mode, the interfaces heuristically indicate the places where the structure may bend easily in a conformational change [22]. In one of the interfaces, we introduce a catalytic core defined as a pair of nodes where ATP can bind (in Figure 2c, the ATP and its binding springs to the nodes of the core are shown in black), one node in a distal cluster and the other in the central one. As many pairs of vertices may exist, a heuristic is applied to select one of them. The docking site associated to the catalytic core is defined as the nodes of the associated distal cluster (Figure 2c, white nodes). Finally, the template is constructed by joining two instances of the structure (one of them the mirror image of the other) at the level of the first vertex in the chain of vertices, and setting a motor head at the opposite end of the structure (Figure 2d). This is inspired in the fact that many molecular motors function as dimers [15], i.e., they are composed of two joined identical proteins, each equipped with a motor head at their other extremity.

B. Evolutionary search

The genotype-phenotype mapping is direct at the morphological level: the genome *is* the 3D structure. At the functional level, however, the configuration of the motor heads is indirectly encoded by the structure, as described in the previous subsection.

To start an evolutionary optimization, the agents in the initial generation are generated as randomly folded chains of 50 nodes, defining relaxed springs between all neighboring nodes. Then, agents are evaluated in the following simulation: they are placed above a straight filament (made of nodes of the same size as the nodes of the structure), such that both docking sites touch it. One of the motor heads is set in the *sticky* state, while the other is set in the beginning of the *relaxing* state. If the structure and the configuration of the motor heads is adequate, coordinated working cycles (that is to say, their states change in a coordinated and cyclic way). After a preset amount of time passes, the simulation is stopped and the fitness is calculated to be the displacement of the agent's

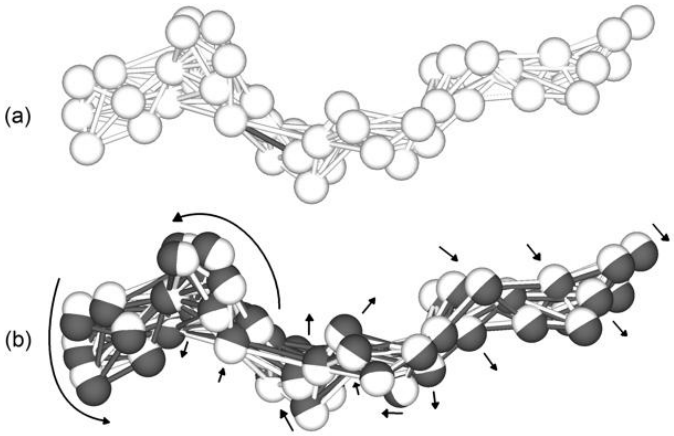

Figure 3. A mass-spring network structure is mutated (a) by enlarging the rest length of a spring (dark gray). The resulting structure after relaxation is shown (b) along with the original structure, in dark gray. Arrows point towards the main direction of displacement in each part of the structure.

center of mass in the direction of the filament, plus the number of completed working cycles by both motor heads.

For some structures, the heuristic cannot properly define the configuration of the motor heads (docking sites and catalytic cores). In this case, they are tagged as *nonevaluable* and are not subject to selection (they are eliminated from the evolutionary competition).

After the evaluation is done, a new population of agents is generated from the previous one by preferentially selecting agents with higher fitness. Finally, the mutation operator is applied (Figure 3). As the genotype-phenotype mapping is direct at the morphological level, the mutation operator must be able to bring many coordinated changes to the structure, in order to be effective. This can be accomplished by using a physics-based mutation: as each network is a spatial configuration of vertices connected by springs in resting state (neither compressed nor stretched), a mutation consists of changing the rest length of one or several springs, each one by an independent, random amount. These perturbations introduce potential energy in the mass-spring network. If it is allowed to relax through a physics simulation, the relative positions of many vertices will change in a coordinated manner (just as originally intended) to relieve the stress. After the relaxation process, the rest lengths of the springs are set to the new distances between nodes, and springs may be added (resp. deleted) if nodes become (resp. cease to be) neighbors. In each evolutionary run, a population of 100 templates undergoes the evaluation-selection-mutation cycle for 200 generations.

III. Results

The model has been tested in 38 evolutionary runs. In each run, 100 random mass-spring networks were generated to compose the corresponding initial population, 3800 in total. Almost all of them either walked a negligible distance or were nonevaluable (Figure 4). However, taking as a reference the distance walked by the best individual in each evolutionary run, significantly improved individuals have evolved, too. In

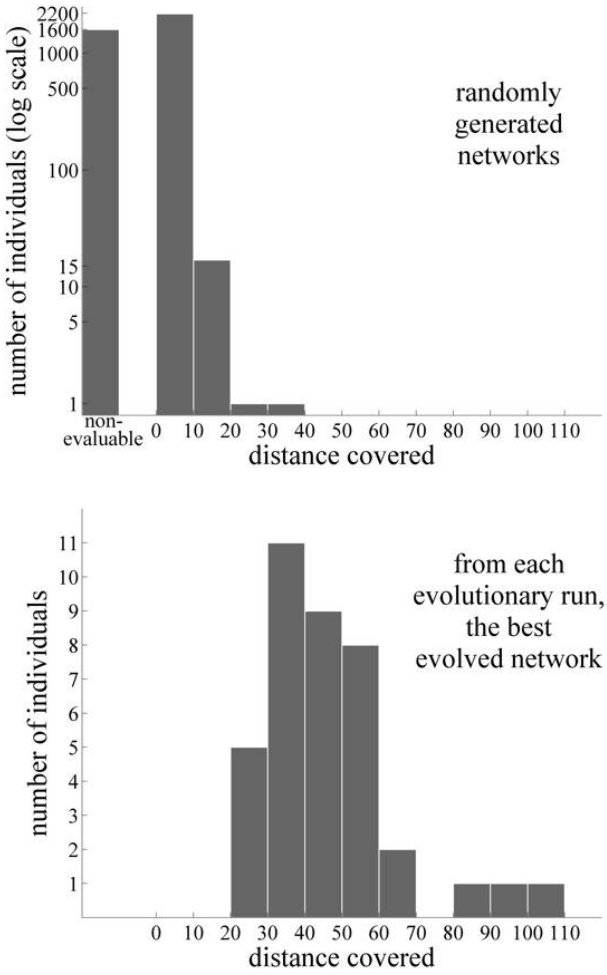

Figure 4. Histograms comparing the performance of 3800 randomly generated templates and the best evolved templates in 38 evolutionary runs. In the first histogram, a significant fraction of the templates ($\simeq$ 1600) are nonevaluable.

many cases, relatively minor modifications to the mass-spring network triggered a significant increase in the distance covered by the corresponding motor templates, suggesting that good templates needed to be precisely tuned to the working cycle and the details of the simulation.

The evolved bipedal templates feature a range of shapes and gaits:

- Walking *pseudo-legs* (Figure 5a) take short and secure alternate steps. The example shown here presents the peculiarity that the legs get attached to the filament at different angles, yet they still produce a steady gait.
- Slow, well-secured *pullers* (Figure 5b) keep a firm grip on the filament. Note that the limbs grasp the filament from below, while they join above it. This example rotates around the filament as it moves along it.
- *Hoppers* (Figure 5c) thrust themselves with both motor heads in an alternate way, only occasionally attaching both legs simultaneously to the filament. In the example provided here, the greater parts of the limbs are entan-

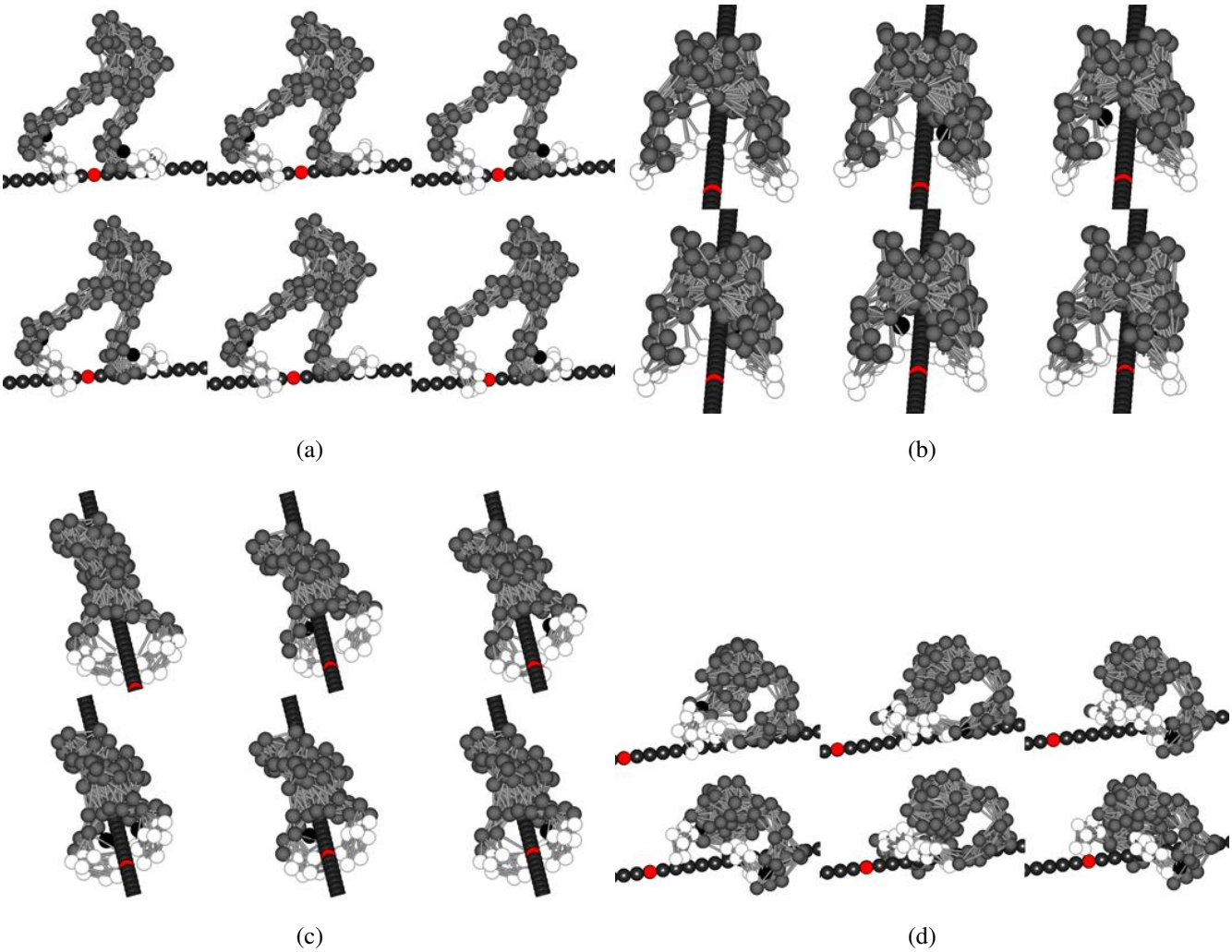

Figure 5. Sequences of snapshots illustrating the gait patterns of four evolved motor templates (in each case from left to right and from top to bottom). A node in the filament is marked in red to provide a point of reference.

gled into a single mass, effectively acting as cargo, and transported by comparatively small actuating limbs.

- Short but fast pulling *pseudo-limbs* (Figure 5d) are the fastest bipedal templates evolved in these experiments. This example has the peculiarity that the phase difference between both legs drifts in time.

IV. Discussion

In this study, we have presented a framework to generate motor templates (walking bipedal agents) inspired by biological molecular motors. The methodology consists of deriving the function of the agents from their structures (based on normal modal analysis), via a simple evolutionary algorithm and a physics-based mutation operator. The resulting structures can be interpreted as models of robotic agents made of elastic materials, suspended in a viscous fluid, while the "ATP molecules" that power the agents can be interpreted as simple actuators modifying the length of isolated parts of the structure.

As the structures are optimized to solve a functional problem (move forward as fast as possible) without morphological specifications, the problem can be described as *behavior-finding* [12] structure or morphology according to a set of constraints. The application of evolutionary optimization to behavior-finding tasks often yields diverse and sometimes unexpected solutions [12].

Many aspects of the model were specifically designed to be as simple as possible. The genome is minimal: it is only a fixed-width sequence of nodes in 3D space with springs between neighboring nodes, and the evolutionary algorithm is also very simple, including a single mutation operator and no crossover. Viable gait patterns could still be found in a high-dimensional space because the search was canalized in two ways:

- The working cycle (a simple reactive model) is hard-wired, and the configuration of the motor heads is indirectly encoded in the morphology of the agent.
- The mutation operator is based on physical relaxation after the application of perturbations to the structure, so it induces a fitness landscape that is more correlated to

the physical characteristics of the structure, which plays a key role in the configuration of gait patterns.

However, these features of the model are relatively low-level and did not constrain in any precise way the gait patterns of the templates. Thus the diversity of shapes and gait patterns was only enabled, not determined, by these characteristics and by the fact that the individuals competed in a 3D virtual world, coevolving their morphologies and behaviors (gait patterns). Morphogenesis arose by repeated application of a complex mutation operator through evolutionary time, instead of leveraging a complex genotype-phenotype mapping. As an example of morphological computation, gaits lacked any specific control subsystem: gait patterns emerged from the interaction between the properties, the physics, and the geometry of the templates and filament.

The mutation operator can also be considered as a mode of morphological computation. Instead of using heuristics based on the analysis of the characteristics of the structures, the mutation operator only perturbed the rest length of one or more springs in the structure. The new structure was then calculated by simulating physical relaxation, which naturally induced many coordinated changes into the mutated structure.

REFERENCES

[1] K. Stanley and R. Miikkulainen, "A taxonomy for artificial embryogeny," *Artificial Life*, vol. 9, no. 2, pp. 93–130, 2003.

[2] G. S. Hornby and J. B. Pollack, "Creating high-level components with a generative representation for body-brain evolution," *Artificial Life*, vol. 8, no. 3, pp. 223–246, 2002.

[3] D. Floreano and L. Keller, "Evolution of adaptive behaviour in robots by means of Darwinian selection," *PLoS Biology*, vol. 8, no. 1, p. e1000292, 01 2010.

[4] J. C. Bongard and R. Pfeifer, "Repeated structure and dissociation of genotypic and phenotypic complexity in artificial ontogeny," in *Proceedings of the 3rd Genetic and Evolutionary Computation Conference (GECCO)*. Morgan Kaufmann, 2001, pp. 829–836.

[5] G. S. Hornby, H. Lipson, and J. B. Pollack, "Generative representations for the automated design of modular physical robots," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 4, pp. 703–719, 2003.

[6] M. Komosinski and A. Rotaru-Varga, "Comparison of different genotype encodings for simulated 3D agents," *Artificial Life*, vol. 7, no. 4, pp. 395–418, Fall 2001.

[7] J. B. Pollack, H. Lipson, G. Hornby, and P. Funes, "Three generations of automatically designed robots," *Artificial Life*, vol. 7, pp. 215–223, June 2001.

[8] L. Spector, J. Klein, and M. Feinstein, "Division blocks and the open-ended evolution of development, form, and behavior," in *Proceedings of the 9th Genetic and Evolutionary Computation Conference (GECCO)*. ACM, 2007, pp. 316–323.

[9] Y.-S. Shim and C.-H. Kim, "Generating flying creatures using body-brain co-evolution," in *Proceedings of the Symposium on Computer Animation (part of the 30th SIGGRAPH)*, ser. SCA '03. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2003, pp. 276–285.

[10] C. Paul, "Morphological computation: a basis for the analysis of morphology and control requirements," *Robotics and Autonomous Systems*, vol. 54, no. 8, pp. 619–630, 2006.

[11] K. Matsushita, M. Lungarella, C. Paul, and H. Yokoi, "Locomoting with less computation but more morphology," in *Proceedings of the 22nd IEEE International Conference on Robotics and Automation.*, April 2005, pp. 2008–2013.

[12] D. Lobo, "Evolutionary development based on genetic regulatory models for behavior-finding," Ph.D. dissertation, Universidad de Malaga, 2010.

[13] R. Pfeifer, F. Iida, and G. Gomez, "Morphological computation for adaptive behavior and cognition," *International Congress Series*, vol. 1291, pp. 22–29, June 2006.

[14] A. J. Rader, C. Chennubhotla, L.-W. Yang, and I. Bahar, *The Gaussian Network Model: theory and applications*. Chapman & Hall/CRC, 2006, ch. 3, pp. 41–63.

[15] M. Schliwa and G. Woehlke, "Molecular motors," *Nature*, vol. 422, no. 6933, pp. 759–765, April 2003.

[16] R. D. Vale and R. A. Milligan, "The way things move: looking under the hood of molecular motor proteins," *Science*, vol. 288, no. 5463, pp. 88–95, April 2000.

[17] J. D. Fernández and F. J. Vico, "Automating the search of molecular motor templates by evolutionary methods," *Biosystems*, vol. 106, pp. 82–93, 2011.

[18] Y. Togashi and A. S. Mikhailov, "Nonlinear relaxation dynamics in elastic networks and design principles of molecular machines," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, no. 21, pp. 8697–8702, May 2007.

[19] M. Lu, "The role of shape in determining molecular motions," *Biophysical Journal*, vol. 89, no. 4, pp. 2395–2401, October 2005.

[20] J. Su, "Protein unfolding behavior studied by elastic network model," *Biophysical Journal*, vol. 94, no. 12, pp. 4586–4596, June 2008.

[21] S. Kundu, D. C. Sorensen, G. N. Phillips, and Jr, "Automatic domain decomposition of proteins by a Gaussian Network Model," *Proteins: Structure, Function, and Bioinformatics*, vol. 57, no. 4, pp. 725–733, December 2004.

[22] L.-W. W. Yang and I. Bahar, "Coupling between catalytic site and collective dynamics: a requirement for mechanochemical activity of enzymes." *Structure*, vol. 13, no. 6, pp. 893–904, June 2005.