What Is Morphological Computation? On How the Body Contributes to Cognition and Control

Vincent C. Müller[*,**]
Anatolia College/ACT
University of Leeds

Matej Hoffmann[†]
Istituto Italiano di Tecnologia
Czech Technical University in Prague

Keywords
Body, cognition, computation, control, embodiment, soft robotics

Abstract The contribution of the body to cognition and control in natural and artificial agents is increasingly described as "offloading computation from the brain to the body," where the body is said to perform "morphological computation." Our investigation of four characteristic cases of morphological computation in animals and robots shows that the "offloading" perspective is misleading. Actually, the contribution of body morphology to cognition and control is rarely computational, in any useful sense of the word. We thus distinguish (1) morphology that facilitates control, (2) morphology that facilitates perception, and the rare cases of (3) morphological computation proper, such as reservoir computing, where the body is actually used for computation. This result contributes to the understanding of the relation between embodiment and computation: The question for robot design and cognitive science is not whether computation is offloaded to the body, but to what extent the body facilitates cognition and control—how it contributes to the overall orchestration of intelligent behavior.

1 Introduction

1.1 Structure

It has become increasingly common to explain the intelligent abilities of natural agents through reference to their bodily structure, their *morphology*, and also to make extended use of this morphology for the engineering of intelligent abilities in artificial agents (e.g., robots). These two uses of morphology for *explanation* and *engineering* are sometimes referred to as "morphological computation" (see the special 2013 issue of *Artificial Life* 19(1)).[1] Since this notion is fairly new and fairly vague, we will look at the existing uses, interpretations, and definitions of this concept and discuss their implications. The term is initially puzzling because it is sometimes used to counter the classical view

---

Author Contributions: The authors are listed in the sequence agreed on when this work was started some time ago. In the meantime the contributions from the roboticist (M.H.) and the philosopher (V.C.M.) have grown to be roughly equal.

[*] Contact author.

[**] Anatolia College/ACT, PO Box 21021, 55510 Pylaia, Greece, and IDEA Centre, Department of Philosophy, University of Leeds, Leeds, LS2 9JT, UK. E-mail: vmueller@act.edu

[†] iCub Facility, Istituto Italiano di Tecnologia, Via Morego 30, 16163 Genova, Italy. E-mail: matej.hoffmann@iit.it; Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Karlovo namesti 13, 121 35 Prague 2, Czech Republic. E-mail: matej.hoffmann@fel.cvut.cz

[1] This morphological computation community unites researchers from various fields, including computer science, robotics, electronics, material science, chemistry, and biology, who are in a broad sense interested in computational properties of different physical substrates or morphologies. Two editions of the *International Conference on Morphological Computation* were held in Venice, Italy, in 2007 and 2011, followed by the *International Workshop on Soft Robotics and Morphological Computation* in Ascona, Switzerland, 2013.

Artificial Life 23: 1–24 (2017) doi:10.1162/ARTL_a_00219

that "cognition is computation" through a study of body-dependent behavior (there it overlaps with embodiment), but then it reintroduces the notion of computation through the back door. It also seems to call processes and features "computational" that would not normally fall under that term—such as the morphology of geckos' feet that allows them to walk on smooth vertical surfaces.

In order to clear up the muddy waters of morphological computation, we categorize the conceptual situation and come to the conclusion that morphological computation is actually extremely rare. We first introduce four types of characteristic cases that are often cited as examples of morphological computation: (1) passive dynamic walkers, (2) self-stabilizing robots and climbing geckos, (3) insect eyes with distribution of light-sensitive cells optimized for particular tasks, and (4) physical reservoir computing (Section 2). Then, we introduce different notions of computation, abstract and physical (Section 3), and apply them to the characteristic cases (Section 4). Based on this analysis, we classify the case studies into three classes: morphology facilitating control, morphology facilitating perception, and morphological computation. This also serves as a "reality check" for several notions of computing (Section 5). Finally, we explain the conclusion that design and analysis should focus not on computation, but on how morphology facilitates cognition and control—how it contributes to the orchestration of intelligent behavior (Section 6) and how it can be specifically designed to do that—either by evolution in the case of animals, or by engineers. A perfect example of the latter is the growing field of soft robotics.

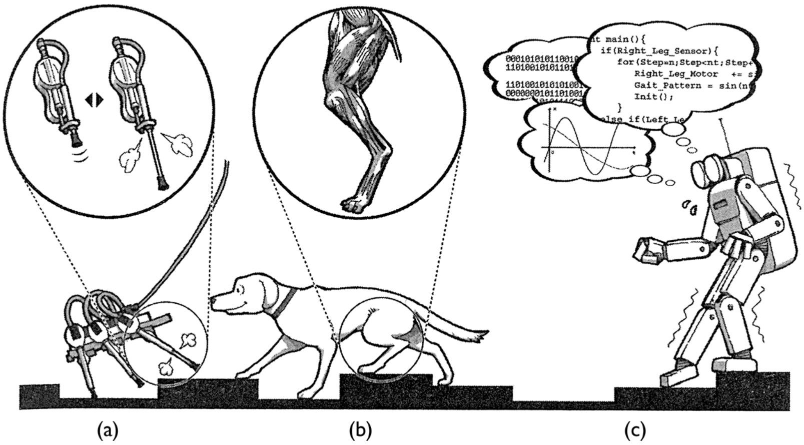
1.2 An Initial Worry

In a classic book on the subject, Pfeifer and Bongard [61] show a flexible sprawl robot, a dog, and a conventional humanoid walking over an uneven surface (see Figure 1) and comment:

> Morphological computation. (a) Sprawl robot exploiting the material properties of its legs for rapid locomotion … thus reducing the need for computation. (b) An animal exploiting the material properties of its legs (the elastic muscle-tendon system) thus also reducing computation. (c) A robot built from stiff materials must apply complex control to adjust to uneven ground and will therefore be very slow. [61, p. 97, Figure 4.1 caption]
>
> By "morphological computation" we mean that certain processes are performed by the body that otherwise would have to be performed by the brain. [61, p. 96]

These remarks lend themselves to at least two divergent interpretations:

1. Appropriate use of the body morphology leads to a reduction of the total amount of computation that is required to complete the task.



```
t main(){
  if(Right_Leg_Sensor){
    for(Step=n;Step<nt;Step+
      Right_Leg_Motor  += s
      Gait_Pattern = sin(n
      Init();
    }
lse if(Left_Le
```

Figure 1. Morphological computation [61, p. 97, Figure 4.1].

2. The total amount of computation is constant, but a body takes over some of the computation originally performed by the brain and computes it in the morphology.

In the example of walking, is this a computational task in the first place? Is the perspective of "offloading computation from the brain to the body" feasible? That is, is it a design decision that one can take: In design (a), one decides to off-load computation to the body, and in design (c), one decides not to, and to compute centrally in the brain/controller? Even if such a hypothetical design landscape existed and an agent could move on it, in what sense would the contribution of the body be a computational one?

2 Approaching the Problem: Characteristic Cases

To clarify the picture, we will analyze four families of case studies from the morphological computation literature and try to structure the diverse approaches by locating them within a conceptual landscape: (1) the passive dynamic walker, (2) self-stabilizing machines, gecko feet, and coffee balloon grippers, (3) the eye of the fly, and (4) physical reservoir computing.

The passive dynamic walker (1) is a classic example that is invoked in many places, but it is also characteristic in that it has no control structure in the classical sense, no computer in the classical sense, no motors, and no sensors either—a purely mechanical device. It is thus the extreme example: If this is computing, what is not? Self-stabilizing machines and gecko feet (2) can be viewed as "active extensions" of the passive walker. The physical body is complemented by actuators and sensors and their connection in a simple control loop; yet the physical interaction of the body with the environment remains absolutely crucial for the task at hand. But the systems discussed in the context of morphological computation concern not only movement, but also traditionally more "cognitive" abilities like perception, and one case where this is particularly apparent, and well-studied, is the eye of the fly (3). Finally, there are cases where the whole system, the body, literally seems to compute, so we need an example for this kind (4), which is *reservoir computing*. Of course, the choice of examples was revised in the light of our discovery that there are three distinct uses of "morphological computing" that actually talk about rather different things—so we need examples for each of these uses. (Cases (1) and (2) belong to the same category, in our analysis.)

2.1 The Passive Dynamic Walker: Behavior by Purely Mechanical Interaction

One classical example in this field is the passive dynamic walker [52]: a minimal robot that can walk without any motors or control electronics. It loosely resembles a human, with two legs, a minimal torso, and two arms, but its ability to walk is exclusively due to the downward slope of the incline on which it walks and the mechanical parameters of the walker (mainly leg segment lengths, mass distribution, and foot shape) (Figure 2a). The walking movement is entirely the result of finely tuned mechanics on the right kind of surface. A motivation for this research is also to show how human walking is possible with minimal energy use and minimal central control.

2.2 Self-Stabilizing Machines, Gecko Feet, and Coffee Balloon Grippers

Most of the problems that animals or robots are faced with in the real world cannot be solved solely by passive interaction of the physical body with the environment. Typically, active involvement by means of muscles/motors is required. Furthermore, the actuation pattern needs to be specified by the agent,[2] and hence a controller of some sort is required. However, it turns out that if the physical interaction of the body with the environment is exploited, the control program can be very simple. For example, the passive dynamic walker can be modified by adding a couple of actuators and

---

[2] In this article, we will use "agent" for whatever acts and has some morphology, like an animal or a robot, without thus committing ourselves to a particular view on what kinds of things are agents, whether they must be individuals, or the like.

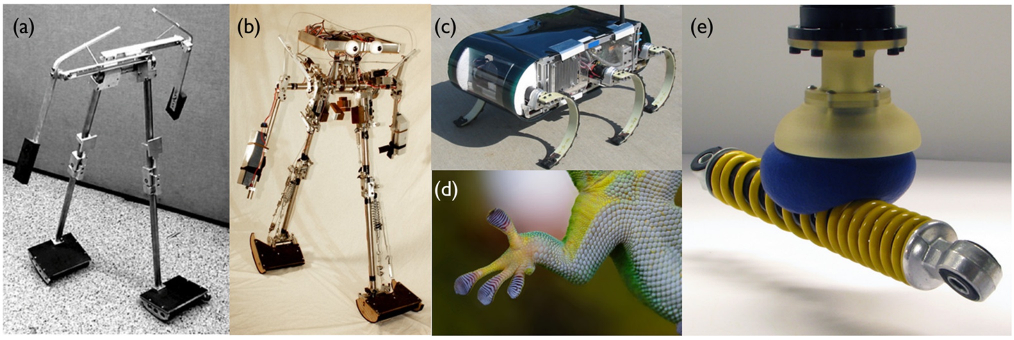


Figure 2. Walking, climbing, and grasping creatures. (a) The Cornell passive dynamic walker with arms, which can walk completely passively down an incline [15]. (b) An actuated extension of the passive walker that can traverse flat ground [14]. (Both pictures courtesy of Steve Collins.) (c) RHex robot [71] (picture courtesy of U. Saranli). (d) Gecko foot. (e) Coffee balloon gripper [8] (picture courtesy of John Amend).

sensors and a reflex-based controller, resulting in the expansion of its ecological niche to level ground (Figure 2b) while keeping control effort and energy expenditure to a minimum [14].

While taking advantage of the natural interaction of the body with the environment can lead to very energy-efficient locomotion, what if the agent is perturbed out of this preferred regime? It seems that corrective action needs to be taken. However, it can be the very same mechanical system that can generate this corrective response. This phenomenon is known as *self-stabilization* and is a result of a mechanical feedback loop. To use the dynamical systems description, certain trajectories (such as walking with a certain gait) have attracting properties, and small perturbations are automatically rejected.[3] Blickhan et al. [5] review self-stabilizing properties of biological muscles in an article entitled "Intelligence by Mechanics"; Koditschek et al. [42] analyze walking insects and derive inspiration for the design of a hexapod robot with unprecedented mobility (RHex—Figure 2c; e.g., [71]). In human locomotion, Taga's seminal work dealing with entrainment among the neural system, the musculoskeletal system, and the environment has a similar spirit [76, 77], including robustness against perturbations.

In some cases, a particular body morphology is the only means to achieve certain behaviors. This is wonderfully illustrated by the ability of geckos to climb both rough and smooth vertical surfaces, mainly thanks to van der Waals forces between their feet and the surface they are climbing on. In order for these forces to have sufficient magnitude, a very intimate contact between the feet and the surface is necessary (Figure 2d). This is achieved through a hierarchical structure of compliance that allows conformation on a centimeter scale (through flexion of body and limbs), millimeter scale (morphology of toes and lamellae on their surface), 1–50-micrometer scale (setae of which the lamellae are composed), and <500-nanometer scale (hundreds of spatulae at the setae tips) [2]. These findings have inspired the design of *Stickybot*—a robot that can climb smooth vertical surfaces including glass, tile, and plastic [39]. Again, the specific ability of the gecko is the result of its morphology interacting with a particular environment—not primarily that of higher-level central control.

Finally, the same equally applies to other behaviors or tasks—not only to locomotion. Brown et al. [8] have devised a gripper that utilizes a unique grasping strategy. Fingers of a hand are replaced by a single mass of granular material (e.g., ground coffee). The bag containing granular material is pressed onto an object, flows around it, and conforms to its shape (Figure 2e). Then, a vacuum pump is used to evacuate air from the gripper, which makes the granular material jam and stabilize

---

[3] The description is idealized—in reality, a walking machine would fall into the class of *hybrid dynamical systems*, where the notions of attractivity and stability are more complicated.

the grasp. The gripper conforms to arbitrary shapes passively, that is, without any sensory feedback, thanks to its morphological properties only. Brown et al. identify three mechanism that contribute to the gripping: (i) geometric constraints from interlocking between gripper and object surfaces; (ii) static friction from normal stresses at contact; and (iii) an additional suction effect, if the gripper membrane can seal off a portion of the object's surface. The properties of the gripper can be changed by using a different granular material. Objects of various shapes as well as hardnesses (from steel springs to raw eggs) can be gripped.

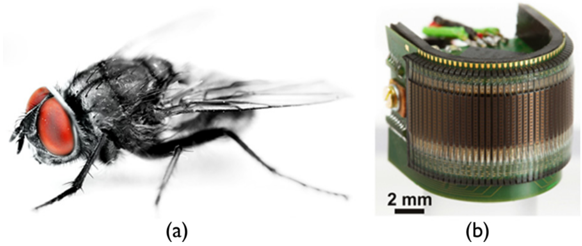
2.3 The Eye of the Fly

In previous subsections, we have outlined a few characteristic cases that illustrate the key role the particular body morphology plays in performing certain behavior. The focus was thus on the implications of the shape and material properties of the body for direct physical interaction with the environment. However, the body morphology critically shapes the information flow in an agent as well. In particular, the type of sensory receptors—their mechanism of transduction—determines what kind of signals the agent's brain or controller will be receiving from the environment. Furthermore, the shape and placement of these sensors will perform an additional transformation of the information that is available in the environment.

For example, different species of insects have evolved different non-homogeneous arrangements of the light-sensitive cells in their eyes, providing an advantageous nonlinear transformation of the input for a particular task. One example is exploiting ego-motion together with motion parallax to gauge distance to objects in the environment and eventually facilitate obstacle avoidance. However, to correctly compute the distance to the objects, the sine of the angle spanned by the object's motion on the eye is involved (see Figure 6 in [25]), which would have to be taken into account by the motion detection circuitry—if the spacing of light-sensitive cells were uniform. However, the distribution of the cells is nonuniform and follows a sine gradient in the interommatidial angle, such that sampling of the visual space is finer towards the front than laterally (the head of a male housefly can be seen in Figure 3a). This effectively compensates for the sine relationship in the formula and allows uniform motion detection circuitry to be used everywhere. Franceschini et al. [25] performed this analysis on a housefly and at the same time designed a mobile robot that avoids obstacles using the same concept. Recent designs of artificial eyes with design inspired by arthropods include [73] and [23] (Figure 3b).

2.4 Physical Reservoir Computing

Unlike typical robot bodies that often have rather simple geometrical forms and are composed of rigid materials, biological bodies typically have highly complex shapes and are soft and deformable. These properties also make their dynamics much richer. This can be exploited for directly achieving particular behavior in the physical world on one hand, and on the other hand can be utilized for computational tasks.



(a) (b)

Figure 3. Eye morphology. (a) Eye of a fly. (b) CurvACE artificial compound eye [23] (image courtesy of Dario Floreano).

The neural networks community has recently proposed a family of architectures that has been named *reservoir computing* (see, e.g., [47] for a review). There is a large collection of neurons with nonlinear activation functions and with recurrent connections that have a random but bounded strength; this is referred to as a *dynamic reservoir*. These neurons are randomly connected to input streams, and the dynamics of the input is then spread around and transformed in the reservoir, where it resonates (or "echoes"—hence the term "echo-state networks") for some time. It turns out that tapping into the reservoir with simple output connections is often sufficient to obtain complex mappings of input stream to output stream that can approximate the input-output behavior of highly complex nonlinear dynamical systems (e.g., [37]). During training, the weights from the input streams and between the reservoir neurons are left intact; only the output weights—from the reservoir to the output layer—are modified by a learning algorithm (e.g., linear regression). The complexity of the training task has been greatly reduced (as opposed to training all the connections—see [36] for details) by exploiting the reservoir to perform a spatiotemporal transformation of the input stream (the temporal aspect of the input sequence has basically been unfolded by the reservoir and can be retrieved directly at any instant). Furthermore, if feedback loops from the output back to the reservoir are introduced and subject to training, the network can be trained to generate desired output streams autonomously.

Interestingly, a physical device can act as a reservoir as well. This can be simply a bucket of water [22] or the body of an agent. Biological bodies interacting with their environments tend to possess the properties required—nonlinearity and fading memory—and might thus be employed as spatiotemporal filters. Hauser et al. [27, 28] provide a theoretical foundation for these mechanisms:

> The underlying idea is to view the morphological structure as some fixed nonlinear *kernel*, which provides us with high-dimensional projections and nonlinear combinations of our input. Hence, the required nonlinearity (next to the dynamics) is provided by the morphological structure itself and, therefore, *linear* feedbacks and readouts are sufficient in order to emulate *nonlinear* differential equations. [28, p. 601]

Concrete demonstrations of this property are provided using simulated mass-spring systems. For example, different networks consisting of dozens of masses and springs with linear feedback loops (see Figure 4) are trained to emulate different nonlinear limit cycles (Van der Pol oscillator, quadratic limit cycle, Lissajous figure) as well as four output streams corresponding to motor patterns for four quadruped gaits. Johnson et al. [38] have demonstrated that a similar mass-spring system can be evolved to replace a continuous-time recurrent neural network in the famous categorization task of Beer [4].

This idea has been adopted by others and demonstrated in other systems too. Caluwaerts et al. [10] have explored this approach in a simulated "robot"—a tensegrity structure consisting of fixed bars and passive and active springs. The body was employed as a reservoir to sense the properties of different terrains. Then, a simple readout mechanism was trained to classify the grounds. Furthermore, the tensegrity structure, accompanied by a feedback loop and a readout mechanism, was also

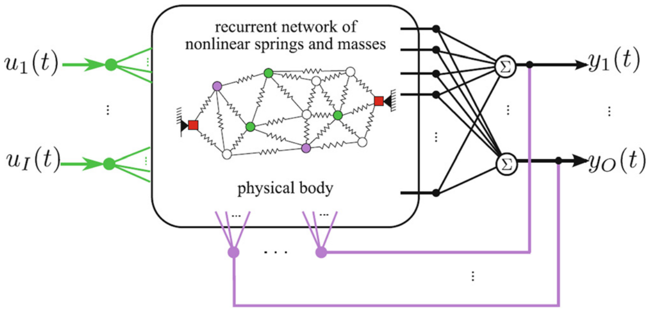


Figure 4. Spring-mass computer [28]. (Image courtesy of Helmut Hauser.)

shown to generate a periodic pattern similar to one produced by a nonlinear oscillator [11]. This output could then be used as a command sequence sent to actuators. However, this approach was further complicated in a quadrupedal robot [19]. Nakajima et al. [57] have used a model of a soft robotic arm inspired by the octopus. In a simulated arm in an underwater environment, they successfully embedded the same benchmarks used by Hauser et al. in the open-loop system (2nd- and 10th-order dynamical system and Volterra series [27]) and in the system with feedback (Van der Pol equations, quadratic limit cycle, and Lissajous curve [28]). Extensions on an actual physical silicone arm in a water tank were presented in [58].

3 The Relevant Notions of Computation: Abstract and Physical

Before we try to analyze the characteristic cases with respect to their relationship to computation, we will set the ground for this exercise by reviewing different notions of computation and, in particular, what it means for a physical system to compute.

3.1 Abstract Computation and Its Models

We will briefly introduce the most important models of abstract or mathematical computation. The first will be the classical Turing model. Second, we will introduce a class of "nonstandard" or "natural" computation models that try to go beyond the Turing model and lift some of its constraints.

3.1.1 Digital Turing Computation

The classical notion of computation starts from a digitally encoded input, which is processed step by step, following an algorithm, giving rise to a digital output. One formalization of this notion is the Turing machine—an abstract device that manipulates symbols on a tape according to a table of rules. The computation is serial and *batch*, that is, the computation is finished and complete output is provided only once the machine halts. No additional inputs can be introduced during the computation. This can be used to define the family of effectively (or algorithmically) computable functions, as outlined by the Church-Turing thesis:

> All and only the effectively computable functions can be computed by a Turing machine.
> [13, 80; also see 55]

Alternative, but equivalent, formalizations are possible using lambda calculus [13], recursive functions [40], or other techniques.

To move from one digital state to another in a Turing machine is a step-by-step process that follows a finite rule, that is, an algorithm, which is itself formal. So, the computation is a purely syntactic process that can be realized in several different ways on a physical device: It is thus *multiply realizable*, and since the states are digital, it is *exactly multiply realizable*.

3.1.2 Natural Computation

The Turing model of computation has been extremely powerful and is sometimes regarded as synonymous with computation itself. However, some of the features of the Turing model (digital, batch, serial, etc.) can also be constraining, especially with regard to realizing the computation on substrates that are common in nature. "Natural computing" is an umbrella term that encompasses efforts to characterize computations performed by natural systems—neural computation, for example—as well as realizing computation on alternative hardware, like molecular (or DNA) computing or quantum computing.

Some classes of problems, like the continuous domain, can only be approximated by Turing machines; some, like parallel asynchronous computing, lie outside the Turing model altogether.

However, to qualify as computation, an alternative model is required that provides a formal description of the states and their evolution.

One possible formalism for analogue computing is provided by dynamical systems theory: A dynamical system is governed by its internal dynamics, and it may also receive a stream of input signals and produce a stream of output signals.[4] Crutchfield et al. [16] provide a broader context for computation by complex systems in their introduction to a focus issue of the journal *Chaos*: "Information Processing in Dynamical Systems—Beyond the Digital Hegemony." Dambre et al. [18] have studied these real-time mappings and defined the information-processing capacity of a dynamical system and how it can be inferred, allowing for a comparison of computational properties of a broad class of dynamical systems. Time-invariant systems with fading memory—which often serve as a formalism in reservoir computing (including physical reservoir computing [27])—are one particular class of these systems. Maass et al. [49] present a computational model based on transient dynamics in high-dimensional neural circuits. Unlike in the Turing model, no sequential transitions between discrete states are necessary, yet universal computational power can be achieved.[5]

In theory, natural computation would be a superset of Turing computation, but it is rather a collection of different possible computational models that lift some of the constraining features of the Turing model (digital, batch, serial, etc.; see [20] for a detailed account). An analogue computer uses continuously changing processes and requires some sort of input, a process, and then a measurement of the outcome to a certain degree of accuracy. A classic example is the slide rule; a less mathematical example is the bombsights used in WWII bomber aircraft to determine the optimal time of bomb release to hit a target, given the angle, altitude, and wind speed (taking into account gravity and air drag). Both were replaced by digital computers in the 1960s. If the computational model is analogue, it is not possible to realize it *exactly* in physical machines, and it is thus only *approximately multiply realizable*.

3.2 When Does a Physical System Compute?

The models of computation discussed above are solely on the abstract level; for computation to be performed in the real world, they need to be *realized* on a physical system. A prime example is the standard digital computer, which comes practically close to realizing the (abstract) Turing machine in an electronic device (only the computer's memory is not unbounded like the tape of the Turing machine). This particular physical realization is unproblematic, and it naturally occurs to us that the digital computer performs computation. However, the situation will become more complicated as we discuss whether other physical systems are computing.

3.2.1 Designed, Useful Computation with Encoding and Decoding

A concrete formal framework addressing "When does a physical system compute?" was recently put forth by Horsman et al. [32] in an article with that title. They tackle the question as a relationship between an abstract object (a computation) and a physical object (a computer), which is best explained by using a schematic (Figure 5) to illustrate the correspondences. These correspondences are encapsulated in a theory T. There are then two principal perspectives one can take. The first one is that of an experimental science, like physics, that aims at understanding a (physical) system. This is a modeling endeavor—a theory is sought that allows describing the evolution of the physical system in abstract terms, and it can be tested using experiments. That is, under the particular theory, the state of the physical system, $\mathbf{p}$, is observed and turned into its abstract counterpart, $m_{\mathbf{p}}$, using the representation relation R_T. Then the physical system is left to evolve, and in parallel the abstract layer is also evolved, using the abstract dynamics, C_T, which should parallel the physical dynamics, $\mathbf{H}$. Finally, if the theory is

---

[4] A possible bridge between dynamical systems and digital computation is the symbolic dynamics framework (e.g., [45]).

[5] This architecture, that of *liquid state machines*, was later united with echo-state networks under the "reservoir computing" umbrella.

$$\begin{tikzcd}
\text{abstract} & m_{\mathbf{p}} \arrow[rr, "C_T(m_{\mathbf{p}})"] \arrow[d, shift left, "I_T"] & & m'_{\mathbf{p}} \approx m_{\mathbf{p}'} \arrow[d, shift left, "I_T"] \\
\text{physical} & \mathbf{p} \arrow[u, shift left, "R_T"] \arrow[rr, "\mathbf{H}\,(\mathbf{p})", "\text{(e.g., voltage changes)}"'] & & \mathbf{p}' \arrow[u, shift left, "R_T"]
\end{tikzcd}$$

Figure 5. Full commuting diagram between an abstract and a physical system. The relationship between the abstract and the physical system is established by the theory T. The dynamics of the abstract system (transition between initial and final states) is described by C_T. The initial state of the abstract system is denoted as $m_{\mathbf{p}}$, and its final state as $m'_{\mathbf{p}}$. The physical system's dynamics is $\mathbf{H}$, and its initial state and final state are denoted as $\mathbf{p}$ and $\mathbf{p}'$, respectively. R_T is the modeling relation that maps physical entities $\mathbf{p}$ to their abstract models $m_{\mathbf{p}}$ (also decoding). I_T is the inverse instantiation relation (encoding) that goes in the opposite direction and allows one to physically instantiate an abstract entity. (Created based on multiple schematics from [32].)

valid, the abstract final state, $m'_{\mathbf{p}}$, should be "close enough" to the representation of the physical final state, $R_T(\mathbf{p}')$, that is, $m_{\mathbf{p}'}$. The abstract and the physical layer should thus "commute," and the theory can be used to predict the evolution of the physical system (running only the path from $\mathbf{p}$ over $m_{\mathbf{p}}$ to $m'_{\mathbf{p}}$).

The other perspective is essentially complementary and involves using the physical system to predict the outcome of an abstract evolution. It relies on the quality of the theory T that produces a commuting diagram, but requires in addition a representation relation in the opposite direction—the instantiation I_T. That is, it has to be possible to find a physical object corresponding to an abstract description (*encoding* the abstract state—e.g., encoding a "1" as voltage high). Here we are moving from the realm of science (modeling) to the realm of technology—the theory can be used to construct (engineer) a physical system in a given state. With that in place, the physical system can finally be used as a computer using the following trajectory: An abstract initial state $m_{\mathbf{p}}$ is encoded into the physical system using $I_T(m_{\mathbf{p}})$, giving rise to $\mathbf{p}$; the physical system is left to evolve, reaching $\mathbf{p}'$, which is finally decoded using $R_T(\mathbf{p}')$, giving rise to $m_{\mathbf{p}'}$, which we are assuming corresponds to $m'_{\mathbf{p}}$—the desired outcome of the abstract evolution. In this case, the physical system has acted as a computer.

Horsman et al. [32] conclude their framework by stating the *necessary requirements for a physical system to be capable of being used as a computer*:

a. A theory T of the physical computational device that has been tested in relevant situations and about which we are confident.

b. A representation $\{R_T, I_T\}$ of the physical system that is used for representing the initial state of the physical system (encoding using I_T) and also for the final state, so that output is produced from the computation (decoding using R_T).

c. At least one fundamental physical computational operation that takes input states to output states.

d. The theory, representation, and fundamental operation(s) satisfy the relevant sequence of commuting diagrams.

This description is agnostic with respect to the particular abstract computation type (digital Turing or natural) as well as to the nature of the physical system—the important thing is that the theory is sufficiently advanced and reliable that we can rely on the commuting diagrams in all scenarios we are planning to use the computational device in. The advantage of a digital computer is that a perfect, rather than a "close enough," match between $m'_{\mathbf{p}}$ and $m_{\mathbf{p}'}$ is guaranteed.

3.2.2 Without the User, There Is No Computer

There is an additional important aspect that is implied by condition b—*representation*, involving encoding and decoding. As Horsman et al. put it:

> A necessary condition of there being representation present is that there is, as well as the computer, an entity capable of establishing a representation relation. That is, an entity that represents this specific physical system as this specific abstract object, encoding and decoding data into it. Something must always be present that is capable of encoding and decoding: if there is a computer, what is using it? [32, p. 15]

This definition matches with the conditions for computation presented by Crutchfield et al. [16, p. 2], who characterize systems performing "designed computation" as machines that "process information in ways that are *useful* to us"—and it is one approach to the symbol grounding problem [54]. In the same vein, Piccinini [66, p. 740] introduces a "Usability Constraint: if a physical process is a computation, it can be used by a finite observer to obtain the desired values of a function"—so he allows for a computer that is not *actually* used—as does the "programmability" proposal in Zenil [84].

Finally, another definition of computation—specifically targeted to encompass morphological computation—was developed for the 2007 International Conference on Morphological Computation under the leadership of Norman Packard (cited from the call for papers of the 2011 conference), defining a process to be a computation if:

(1) we can identify relation between input and output systematically,
(2) the process is programmable for the generation of different classes of non-trivial input-output mappings, and
(3) it is useful for some specific purposes.

Again, conditions (2) and (3) seem to step out of the computational process itself and take an outside perspective of an observer or user of the computation. That is, the mappings need to be programmable (*by someone*) and useful (*for someone*). This observer or user can presumably be the agent itself, or it can be humans that exploit a particular physical system to perform a specific type of mapping that is useful to them. The encoding/decoding step is not explicitly addressed here, however.

3.2.3 Intrinsic Computation

The world is naturally populated with a variety of physical systems of different properties and scales. Many of them can be said to have some computational properties (like storing and generating information) without being "designed computers," so they "intrinsically compute" [16]. Information-theoretic methods can be used to measure the amount of computation (see, e.g., [83] for an example in quantum systems). Some physical systems will then have richer or more interesting computational properties than others—for example, they can be very fast or compute in parallel. Specifically relevant to morphological computation, biological bodies interacting with their environments tend to possess nonlinearity and fading memory, which can be exploited by the reservoir computing framework (Section 2.4).

If there is a user, encoding and decoding are designed, and the physical system is sufficiently reliable, then the natural computing substrates can become computers according to the framework presented in the previous section. However, Horsman et al. [32] point out that this computational (abstract) description of the physical evolution should be available a priori. Only then can the physical system be used to *predict* anything (the evolution of an abstract system). Conversely:

> A common, and unfortunate, method of ascribing computational ability to a non-standard system is as follows. A novel computing substrate is proposed (a stone, a soap bubble,

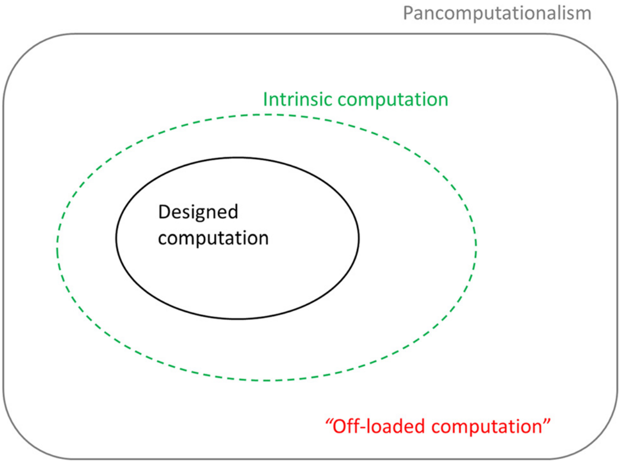


Figure 6. Different notions of computation performed by a physical system. Designed computation refers to physical systems engineered in such a way that their physical evolution can be used to predict the outcome of an abstract evolution (see Section 3.2.1). Intrinsic computation lacks a proper definition, but refers to natural physical systems with interesting computational properties (Section 3.2.3). Pancomputationalism spans the whole hypothetical space, claiming that everything computes (Section 3.2.4). Finally, “offloaded computation” is defined negatively as computation that does not have to be performed by a brain/controller (Section 3.2.5).

> a large interacting condensed-matter system, etc.). The physical substrate is “set going” and an evolution occurs. At a certain point, the end of the process is declared and measurements taken. The initial and final states of the system are compared, and then a computation and a representation picked such that if the initial state and final states are represented in such a way, then such a computation would abstractly connect them. The system is then declared to have performed such a computation: the stone has evaluated the gravitational constant, the soap bubble has solved a complex optimization problem and so on.… If a computational description of a physical evolution can only be applied post-hoc, then the system has not acted as a computer. [32, p. 20]

They argue that numerous representations can be invented post hoc, and hence this approach may lead to a special type of pancomputationalism (“everything computes”—see Section 3.2.4 below). Intrinsic computation is thus loosely defined, encompassing all physical systems with some “interesting” computational properties. We have depicted this schematically in Figure 6 as a superset of designed computation.

3.2.4 Pancomputationalism: Everything Is a Computer

The position that the universe is a computer, that all processes are computation, is now often called “pancomputationalism,” a term probably first used by [65] and [24, p. 566]. In its strongest form, pancomputationalism claims that the universe is literally a computer that computes the changes of the universe according to the physical laws; that all processes are computational processes. Our conventional computing machines would thus be only a tiny fraction of all the computing going on. This is really a cosmological thesis of physics and has its origins in that discipline. For a discussion and recent literature, see [21, esp. Section 2; 56]. Pancomputationalism implies that cognitive and morphological processes are trivially computational in the sense that everything else is.

In the morphological computation literature, the pancomputationalist dissolution looms in remarks like “Control implies computation” (Füchslin et al. [26, p. 10]). If one takes this literally, then no difference between computational and other control can be made any more. But we normally want to say that, for example, the centrifugal governor in a steam engine is a dynamic system and provides proportional control. The centrifugal governor is not a computer, even if it can be replaced by a computational system that measures the relevant values, computes a response, and controls the relevant value (valve opening). If the remark "this is a computational process" deteriorates into "this is a process," then it becomes pointless. In Figure 6, this is simply illustrated by pancomputationalism being the ultimate superset.

3.2.5 Morphological Computation as Offloading from the Brain

A notion of computation that is very popular among the morphological computation community seems to rely on a negative or complementary definition of morphological computation. This is computation in the sense of offloading from brain to body (e.g., [61, p. 96]) that we introduced in Section 1.2. Another instance appears in Hauser et al. [27, p. 356]:

> A rigorous implementation of this concept [morphological computation] are passive walkers. The first of a series was developed by McGeer [52]. Typically, such a robot has no active controller at all. Only its passive physical structure maintains the balance in a robust fashion, while it walks down a slope. Therefore, one could argue that the computation, which is needed in order to balance the robot robustly, is "computed" by the physical body itself.

The offloading argument is also picked up in Füchslin et al. [26] and illustrated in Figure 1 there. Interestingly, the schematic matches very well with our Figure 5. Therefore, we have adapted our Figure 5 to match the one from [26], resulting in our Figure 7. The goal is to bring a physical system (e.g., body of an agent) from initial state $\mathbf{p}$ to final state $\mathbf{p}'$. This is achieved jointly by the physical dynamics $\mathbf{H}$ and by the action of the controller that processes the sensory input to obtain a representation $m_{\mathbf{p}}$ of the physical system's state $\mathbf{p}$. With a goal state $m'_{\mathbf{p}}$, in mind, it computes a control action that should bring about the desired transition and translates this into actuator commands. At the same time, the physical system is subject to its own evolution (dynamics), $\mathbf{H}$. In passive walkers, the abstract level is completely absent; there is only physical dynamics. Conversely, traditional robotics tends to suppress "body dynamics" or "natural dynamics" (see, e.g., [33] or [12] for a formal definition) in favor of bringing about the desired state transitions by means of the computed control action. The idea of "morphological control" is to exploit the physical dynamics to the maximum, while at the same time reducing the contribution of the controller (the abstract-level loop)—as illustrated by the bold arrow $\mathbf{H(p)}$ in the schematic.[6] This is then interpreted as offloading computation from the brain/controller to the body.

It may be possible to examine the amount of computation on the controller side, and different controllers can then be compared. This exercise becomes more interesting if one is allowed to modify also the physical system (the agent's body), although this then becomes much harder to compare formally. Rückert and Neumann [70] studied learning of optimal control policies for a simulated four-link pendulum that needs to maintain balance in the presence of disturbances. The morphology (link lengths and joint friction and stiffness) was manipulated, and controllers were learned for every new morphology. They showed that: (1) for a single controller, the complexity of the control (as measured by the variance of control gains of a time-varying linear feedback controller) varies with the properties of the morphology: certain morphologies can be controlled with simple controllers; (2) optimal morphology depends on the controller used; (3) more complex (time-varying) controllers achieve much higher performance than simple control across morphologies.

Among those conclusions of Rückert and Neumann [70], (1) could be interpreted in the "offloading" spirit, but the performance really depends on a complex interplay of the controller, body, and environment, and the offloading or outsourcing will thus practically be very complex and hard to express in a formal framework. More importantly, this scenario allows us to understand whether the contribution of the body is computational in the sense of the notions presented above.

---

[6] Some schemes from control theory with appropriate cost functions will automatically result in minimal control actions and thus "optimize the contribution of the morphology." For example, Moore et al. [53] used discrete mechanics and optimal control to steer a satellite while exploiting its dynamics to the maximum.

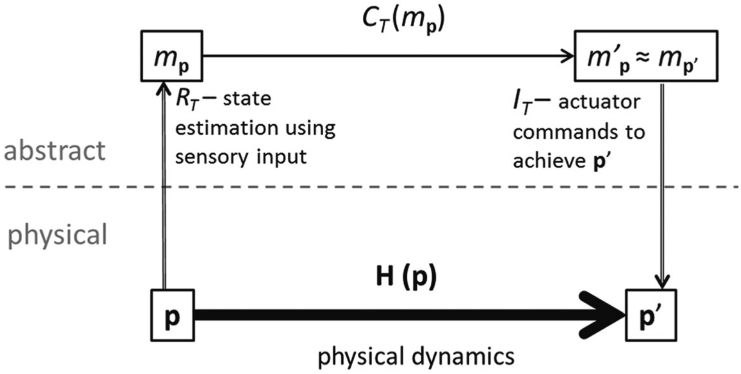

$C_T(m_p)$
m_p
$m'_p \approx m_{p'}$
R_T – state estimation using sensory input
I_T – actuator commands to achieve **p′**
abstract
physical
H (p)
p
p′
physical dynamics

Figure 7. Exploiting morphology for control. See text for details, and also Figure 5 for the definition of symbols.

We claim that it is clearly not, in any sense but the pancomputationalist one. Adopting the framework of "designed, useful computation with encoding and decoding" (Section 3.2.1), we see that the physical system is definitely not used to predict any abstract evolution. Instead, the goal is in the physical space (such as walking), and the abstract level may only serve to help achieve this in the form of a controller (in fact, the abstract computation performed by the controller will have to run on some hardware too, but this will normally not be the agent's body that is walking, but a neural circuit or a digital chip). We are not interested in computational properties of the physical system either (what the passive walker or the pendulum is intrinsically computing). Therefore, the label "morphological computation" is misleading, as the body is not computing in any but a metaphorical sense of taking away some load from the brain—perhaps the label "morphological control" might be appropriate [26].

Thus, faithful to the notion of a complement to brain computation, one can illustrate this offloaded computation schematically in Figure 6 by interpreting the physical computation types we have listed so far as sets. Then, if one assumes that there is a task (say walking) and that all processes are computational (pancomputationalism), the amount of computation needed for the task forms a universe P. Then, if one knows the amount of computation performed by the central controller, say C, then the morphological computation in the outsourced sense could be the absolute complement of the set that represents computation performed by the central controller, that is, P excluding C. The offloaded computation, O, would be equal to

$$O = (C)^{\mathrm{C}} = P \backslash C.$$

This is schematically illustrated in Figure 6.

3.3 Computing Concluded

We have reviewed different types of abstract computational models as well as a number of perspectives on when a physical system computes. For our purposes—assessing when physical bodies act as computers—it is the latter that is key. To this end, we find the first notion presented, "designed, useful computation with encoding and decoding" (Section 3.2.1), the most concrete and strict and also the most useful one. The second one, "intrinsic computation" (Section 3.2.3) is more problematic, but if a system is found to have "interesting intrinsic computational properties," it can be further inspected (if it's a natural system) or modified (if it's artificial) to see if the stronger requirements—encoding, decoding, user—can be established. Paraphrasing Horsman et al. [32, p. 15], this allows us to distinguish between a physical system "going about its business" while *potentially* being a computer, and a system whose physical evolution is used to compute. The pancomputationalist notion of computation (Section 3.2.4) does not seem to be helpful to our analysis. Finally, the computation in the sense of offloading from brain to body (Section 3.2.5) seems to be also a problematic perspective that is hard to defend beyond the level of metaphor.

4 A Look at the Characteristic Cases: Computation?

Let us now look back at the characteristic cases that we introduced in Section 2 in the light of the different notions of computation. We summarize our findings in Figure 8.

4.1 The Passive Dynamic Walker

Although the passive dynamic walker (introduced in Section 2.1) is often taken as a prominent example of morphological computation, it really is only "pure physics walking" or a "physical system going about its business." That is, it definitely does not satisfy the necessary conditions for "designed, useful computation with encoding and decoding" (Section 3.2.1). There is no abstract computation that it was designed to predict through its physical evolution (walking down the slope). Also, although McGeer [52] does offer a model of the walker—its equations of motion—it would hardly qualify as a system with particularly interesting computational properties and as thus computing intrinsically (Section 3.2.3). Therefore, it is hard to conceive of the walker as computing in any but the pancomputationalist sense, that is, the sense that all physics is computing. Finally, is the walker doing computation that was offloaded from the brain? Hardly, as there is no brain the computation of which could be offloaded. At the most, the hypothetical computation that "is needed" for walking has been fully offloaded from a hypothetical controller to the morphology of the walker. Yet, this offloading landscape is purely metaphorical. To illustrate this graphically, we have simply located the passive walker inside the pancomputationalist universe in Figure 8, possibly qualifying as the offloaded computation showcase, but, again, only in a metaphorical sense.

4.2 Creatures Exploiting Morphology to Move

A similar line of argumentation holds for the other examples introduced in Section 2.2. The active descendants of the passive dynamic walkers (Figure 2b), the self-stabilizing robots (e.g., RHex—Figure 2c), the gecko (Figure 2d), or the coffee balloon gripper (Figure 2e) do not seem to be computing in any but the trivial, pancomputationalist, sense either. They may be more useful than the passive dynamic walker, but it is directly their physical behavior in the environment that is useful—there is again no abstract computational level that would, under some encoding and decoding, "commute" with the physical level. All of them have a brain or controller of some sort, so the

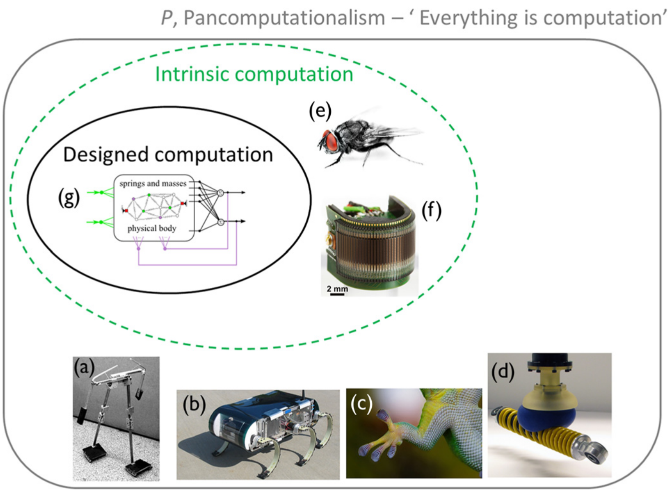


Figure 8. Characteristic cases and the notion of computation involved. (a) Cornell passive dynamic walker [15]. (b) RHex robot [71]. (c) Gecko foot. (d) Coffee balloon gripper [8]. (e) Eye of the fly. (f) Artificial compound eye [23]. (g) Spring-mass computer [28].

offloading thought experiment is somewhat facilitated. One can imagine this in the RHex robot, for example: If the body had a different design, a more complicated controller might be needed to stabilize the robot during locomotion. However, the gecko feet seem to further compromise the offloading perspective—if the gecko did not have feet with the particular morphology, it would simply fall off the vertical surface, no matter how big a brain was available. In conclusion, we think that all these are examples of *morphology facilitating control* and that a computational perspective is misleading. We will devote a separate subsection (Section 5.1) to these cases.

4.3 The Insect Eye Preprocessor

Unlike in the previous cases, a computational perspective seems better applicable to the eye of a fly (Section 2.3). The eye morphology performs a nonlinear transformation of the visual input that greatly facilitates subsequent processing. Does this case fit the framework of Horsman et al. [32]—Figure 5? Not really. The problem that needs to be solved by the agent is not solely an abstract one, but lies at the interface of the physical world, which needs to be sensed, and the abstract world, where the agent needs to estimate the distances to objects.[7] That is, the agent does not start from an abstract problem it would need to solve by encoding it into a physical system and then reading it out again after the physical system's evolution. Instead, objects in the physical world are transformed into activations of light-sensitive cells and later into neural activations ("decoded" from the physical to the abstract system in the sense of Figure 5—this corresponding "encoding" according to standard usage in nueroscience though). After that, this information is processed by the agent's brain to generate an appropriate motor response (obstacle avoidance). It is a sensory-motor loop that brings about control of the flying agent, and it is the first part of the control loop where the morphology steps in. There, part of the transformation that needs to be accomplished is carried out by the morphology itself, leaving less work to the neural circuitry after processing of the sensation takes place. One could even say that evolution has "designed" the morphology to compute this function. Nevertheless, the whole loop does not match with the one that corresponds to using a physical system to predict abstract evolution, as required by the strict definition presented in Section 3.2.1. On the other hand, one could say that the eye morphology has interesting computational properties (equivalent to a function involving a sine term), which are "intrinsically" performed by the morphology. This is illustrated schematically in Figure 8.

4.4 Computing with Masses and Springs, Humanlike Bodies, and Octopus Arms?

Buckets of water, masses and springs, tensegrities, and octopus arms were introduced earlier (Section 2.4). Interestingly, they all satisfy the properties that are necessary to act as a reservoir: high dimensionality, nonlinearity, and fading memory. They can thus be employed as a nonlinear kernel to perform a spatiotemporal transformation of incoming data. In a second stage, using a readout mechanism implemented in a "classical" computer, they give rise to a powerful information-processing pipeline, which is particularly suited for some tasks, like classification and prediction of time series. In a biological context, perception is a prime example where classification and prediction of incoming input streams is highly relevant. In this sense, the biological reservoirs—such as soft bodies—perform a similar function to that of the eye morphology from the previous section. First, however, they add the temporal dimension to the sensory preprocessing they can perform. Second, unlike the insect eye, which is specialized in performing one particular transformation, the spatiotemporal unfolding of the input stream performed by the reservoir is more universal, and there can be several readout mechanisms that take advantage of it.

The computational capabilities of a reservoir can be further increased if the loop is closed by feeding the reservoir's output back to it. Then, it is possible to autonomously generate a broad class of limit cycles, such as the Van der Pol oscillator, quadratic limit cycle, and Lissajous figures [28]. Importantly,

---

[7] This division—abstract versus physical—is quite arbitrary, though. In the organism, this sensing task is part of a closed loop that is connected to the motor neurons in the fly's wings that directly bring about the obstacle avoidance.

these have been demonstrated in two different physical (simulated) reservoir computing setups: a mass-spring system [28] and an octopus arm [57].

With this additional step, it seems that the reservoir is truly used as a computer—it will compute the evolution of an abstract mathematical system, such as a nonlinear oscillator. The model of abstract computation is likely to be one of the natural computation class (Section 3.1.2). For example, Hauser et al. [28, p. 356] write: "Regarding the type of computation, we consider mathematical models, which can be characterized as complex mappings of input to output streams in continuous time." In this case, we are thus for the first time dealing with a proper computer in the sense of the requirements of "designed, useful computation with encoding and decoding" (Section 3.2.1) (hence its placement in Figure 8), with perhaps one reservation: Although the bodies can be recruited for computation, one can hardly say that they were "designed" with this goal in mind. Furthermore, since different dynamical systems benchmarks were instantiated in different reservoir systems, there is even an important additional feature of computation—that of *universal computation*. It thus follows automatically that these systems would also be intrinsically computing and they are naturally also compatible with the offloading perspective. In fact, taking the case of biological bodies, one could perhaps say that an additional computing substrate is instantiated in the body where the brain could outsource its tasks.

5 Discussion: Three Roles of Morphology

In our survey of the case studies, we have come across different instances of morphological computation and we have examined to what extent they satisfy their respective requirements. Confronting the characteristic cases with these conditions gave rise to their approximate localization on the landscape of the computational notions (Figure 8). Interestingly, many ended up on the outer circle, far from the imaginary computational core at the center of the schema. In light of this analysis, we propose a simple classification of the case studies into three types:

A. Morphology facilitating control
B. Morphology facilitating perception
C. Morphological computation

This classification, along with some representatives for each category, is depicted schematically in Figure 9. The true computational character of the deployment of morphology increases from left to right.

5.1 Morphology Facilitating Control

This category could also be called "morphological control" [26], "mechanical control," or, in a narrower, open-loop sense, "morphology facilitating actuation." The passive dynamic walker, the creatures exploiting mechanical self-stabilization, the gecko relying on its special feet, and the jamming-based coffee balloon gripper would all fall into this category (Figure 9a–d). Also, all the examples of locomotion (quadruped Puppy and fish Wanda) and grasping (Yokoi hand) cited in [63] belong here. Other fitting representatives would be the pneumatic actuators embedded in elastomers that can move and grip [35], and the switching of behaviors exploiting attractors in the mechanical system in [59]. In all these cases the body is ingeniously contributing to the task; it enables physical behavior in the real world. However, as we argued in Sections 4.1 and 4.2, it is hard to conceive of the systems as computers in any but the pancomputationalist sense. The offloading computation perspective is also problematic, since, first, the possibility of shifting tasks between the controller and the body is practically limited; second, even if through better exploitation of body morphology the controller can be simplified (and that can be quantified), that does not imply that the body has taken over any computation. In fact, calling it computation may allude to a much stronger thesis, namely that what is done by the interaction of the body with the environment here could also be done by a conventional computer. And that is quite the opposite of what most proponents of the morphological stance want to say, since they want to stress the importance of embodiment, in fact its indispensability. Therefore, our claim is that exploiting morphology for

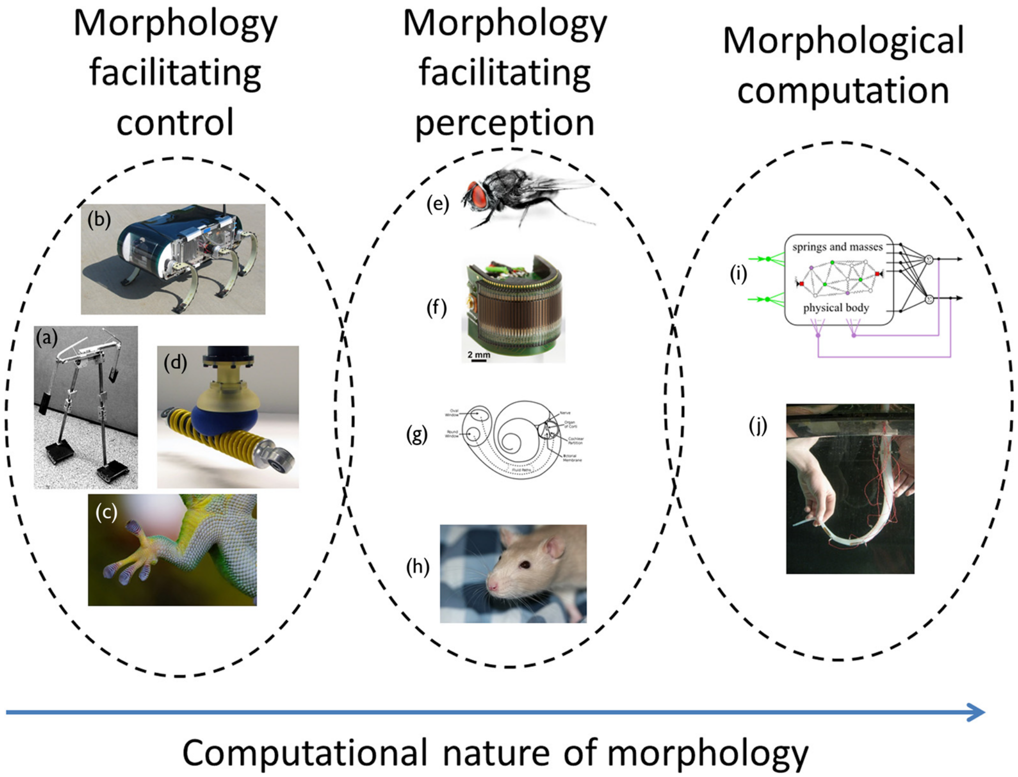

Morphology facilitating control
Morphology facilitating perception
Morphological computation
(b)
(a)
(d)
(c)
(e)
(f)
2 mm
(g)
(h)
(i)
springs and masses
physical body
(j)
Computational nature of morphology

Figure 9. Three roles of morphology. The passive dynamic walker (a) [15], the robot RHex (b) [71], the gecko foot (c), and the coffee balloon gripper (d) [8] are featured as representatives of morphology facilitating control. The eye of the fly (e), artificial compound eye (f) [23], cochlea (g), and whiskers of a rat (h) are examples of morphology facilitating perception. Morphological computation is represented by the spring-mass computer [28] and the octopus arm reservoir [58]. (Picture G is from Wikimedia Commons; picture H by Dawn Huczek, Wikimedia Commons; picture J courtesy of Kohei Nakajima.)

control should be better handled in a non-computational context. This argument is elaborated in [30] along with a discussion of the implications of simple versus complex bodies.

5.2 Morphology Facilitating Perception

The eye-of-the-fly case study (Sections 2.3 and 4.3) is much more compatible with a computational viewpoint. The transformation performed by the insect eye is a well-defined input-output mapping that is useful for the agent and provides input to central processing. In this case, the morphology performs an important preprocessing step—takes over part of the transformation that would otherwise have to be performed by the brain at a later stage. Other examples that might fall into this category are the non-homogeneous distribution of photoreceptors in the human eye (optimized for acuity rather than motion detection), the mechanotransduction in the cochlea of the ear [50], and the sensilla morphology in crayfish [75]. Recent examples from robotics include strain-vector-aided sensorization of soft structures [17] and sensing through compliant artificial pneumatic muscles [34, 74]. However, the morphology (the physical system) is not deployed here as a computational substrate to calculate an abstract problem—this would involve encoding the abstract entity into the morphology and eventual decoding. Instead, the morphology acts at the interface between physical quantities, and their representation in the brain/controller of the agent. At this interface, they introduce advantageous transformations that facilitate subsequent processing. Obviously, the morphology has no ambitions regarding multipurpose or universal mappings; on the contrary, it is highly specialized for one particular transformation.

Of course, the preprocessing performed by the morphology tells only half of the story—a *passive perception* one in this case. Organisms interact with their environments in a closed-loop fashion: Sensory inputs are transformed into motor outputs, which in turn determine what is sensed next. Therefore, additional advantageous transformations can be achieved if *active perception* (e.g., [3]) is considered. Active touch constitutes a perfect example where the sensor morphology, such as the length, placement, and mechanical properties of a rat's vibrissae (whiskers), is as important as the *whisking strategy*—active exploration of the environment (see [67] for a survey). These findings can also be translated into artificial systems, such as whiskerlike sensors [68, 64]. Lungarella and Sporns [48] demonstrate how sensory morphology and closed-loop interaction can be quantified using information-theoretic measures. An example on this theme, where sensor morphology and active sensing go hand in hand and biological inspiration finds its way to robotic systems, is underwater electrical sensing [6, 7]. However, for the purposes of this article, the effect of sensing actively cannot be directly attributed to the morphology anymore and thus has remained essentially out of our scope here.

Four representatives of this category are pictured in Figure 9e–h.

5.3 Morphological Computation

Finally, the only case studies that came close to truly computing were the physical reservoir computing cases (Sections 2.4 and 4.4). Employing the reservoir computing framework, systems that have properties akin to human or animal bodies can be shown to have interesting capabilities in performing advantageous—and quite general—spatiotemporal transformations of incoming time series. Moreover, they are capable of autonomously emulating a broad class of dynamical systems. As this has been demonstrated in multiple systems (masses and springs, octopus arm), these are, in our view, demonstrations of true computation that can be performed by humanlike morphology.

Let us note here that the reservoir computing approach has also been successfully implemented in optoelectronic setups [44, 60]. Nonlinear optoelectronic oscillators subject to delayed feedback have been used to perform digit recognition and time series prediction [44] and speech recognition and nonlinear channel equalization [60], using the reservoir computing concept. Furthermore, a first design for an analogue readout mechanism in an optoelectronic setting has been presented [72], giving rise to a fully analogue setup. However, the optoelectronic implementations have a solely computational goal—they are basically an alternative substrate, in which the combination of physical properties with the reservoir idea gives rise to a practically very powerful computer. Thus, they would fall into the realm of computation performed in alternative hardware, such as molecular or quantum computing, which really is full-blown computer technology where the morphology is designed to be transparent with respect to the computation; such systems have remained largely outside of our scope. These setups work on a completely different scale and speed than the macroscopic human-body-like reservoirs mentioned earlier, which grants them higher bandwidths.

On the other hand, the macroscopic systems—such as nonlinear compliant bodies of human size—have intrinsic limitations with respect to computational tasks. In particular, they are very slow and noisy and they have to perform other tasks at the same time. Yet, they are there for free, so to speak, so the agent may consider employing their computational resources for some tasks, and reservoir computing can provide the framework. This very much blurs the boundaries between the brain as the seat of computing and the "mere physical body," in accordance with biological reality, which will be discussed in the next section. However, thus far the examples of this type [28, 57, 58, 11, 19, 37] have a theoretical or proof-of-concept character, and their applicability remains to be proven (Figure 9I–J).

6 Applications and Conclusion

6.1 Biological Reality Blurs the Boundaries

We hope the classification above is a worthwhile contribution that will help to clear up the muddy waters of morphological computation and at the same time facilitate progress in very practical terms when designing systems. One additional caveat, however, may lie in the fact that a clear separation of the tasks is not always possible. Whereas in artificial systems like robots there is classically a complete separation of the controller (which is a computer) and the body (also called plant—a physical device), in biological systems the brain and the body are so tightly intertwined that this separation is not feasible. The central and the peripheral neural system are often thought of as processing information or as computing. For example, Christoph Koch begins his classic book *Biophysics of Computation* with the remark: "The brain computes! This is accepted as a truism by the majority of neuroscientists engaged in discovering the principles employed in the design and operation of the nervous system" [41, p. 1]. This seems to be a prevalent framework at this point, and we will not question it here. However, the non-neural tissue has many more complex functionalities than being just a muscle-tendon system that is commanded by the brain. Rieffel et al. [69] note that apart from transmitting mechanical properties such as power or balance, the physical forces and displacements are also involved in non-neural conduits of information.

We can perhaps say that the interaction of biological organisms with the environment is orchestrated through a continuum of processes that can roughly be layered from the most low-level and physical, through reflexes and simpler circuitry in the spine, to high-level control consisting of pure information processing with high bandwidths in the brain. All layers are necessary, but the lower the level we want to understand, the less success we will have with a computational approach.

6.2 Amidst Concepts and Buzzwords: From Computation to Orchestration?

Undoubtedly, there is a lot of potential in recruiting the morphology for all the tasks we have described above. Our initial worry (Section 1.2) originated from the tension we perceive within the "morphological computation" label. As we argued, morphology is absolutely essential for a wide range of behaviors that animals or machines are engaged in. However, the contribution of the morphology is largely non-computational. Other concepts or labels have been proposed: intelligence by mechanics [5], morphological control ([26]; but a search suggests that this seems to be a term used often in chemistry, referring to the shapes of different compounds), mechanical control (used in different fields in different contexts), anatomical computation [81], implicit intelligence (abstract of R. Fearing's lecture at ETH Zurich, November 2012), and morphological communication [69]. Furthermore, there is a large overlap with the notion of *embodiment*, which is commonly used to capture the phenomena of our sample cases (cf. [82], [9], and [31], where some of the characteristic cases used here are interpreted as embodiment showcases).

Perhaps one can say that embodiment is typically associated with slightly higher-level phenomena, in particular how the embodied interaction with the environment affects cognition, while the morphological computation community is typically concerned with lower-level traits. Still, as we have argued, morphological computation refers to facilitating control and preprocessing sensory information (thus facilitating perception), as well as to the possibility of "true" computation performed by the body. This is also reflected in the list of buzzwords above—for instance, intelligence by mechanics, morphological control, and mechanical control seem to highlight the exploitation of morphology to simplify a control task, whereas morphological communication seems to emphasize the information flows in the morphology. Perhaps, trying to cover all these facets with a meaningful label is futile. The overarching goal is to orchestrate the morphology so that it contributes as much as possible to all the tasks the agent needs to master. In fact, *orchestration*, or "orchestration of behavior" [63], seems to be an expression appreciated by the community,[8] which does not have any undesired connotations or legacy.

6.3 Conclusion

In our discussion of the increasingly popular notion of morphological computation, a number of well-known case studies that are often cited in this context were confronted with different notions of computation. We concluded that many of the morphological computation showcases, such as

---

[8] As witnessed in the discussion at *International Workshop on Soft Robotics and Morphological Computation* in Ascona, Switzerland, 2013.

passive dynamic walkers, fail to qualify as computation in other than a metaphorical sense. Three categories were proposed: (A) morphology facilitating control, where the walkers and other creatures that exploit their morphology (like geckos) fall; (B) morphology facilitating perception, encompassing mostly case studies where body or sensor morphology aids perceptual tasks; and (C) morphological computation proper, where only physical reservoir computing qualified.

In terms of applications, leaving aside computing on nonstandard hardware, the most relevant area where exploitation of morphology is and will be the key is probably robotics, and in particular soft robotics ([1, 62, 79] and the first issue of the *Soft Robotics Journal* [78]). Soft robots break the traditional separation of control and mechanics and exploit the morphology of the body and properties of materials to assist control as well as perceptual tasks (see also [51] for a recent survey of materials and technologies that can facilitate this). Pfeifer et al. [63] even discuss a new industrial revolution. Appropriate, "cheap" designs lead to simpler control structures, and eventually can lead to technology that is cheap in a monetary sense and thus more likely to affect practical applications. Yet, a lot of research in design, simulation, and fabrication is needed (see [46] for a review). Hermans et al. [29] recently proposed a machine-learning algorithm to address the first challenge—design—if an approximate parametric model of the system's dynamics and examples of desired behavior are available. Kurowski and von Stryk [43, p. 3771] proposed a "systematic approach to the design of embodiment" applied to a compliant legged robot.

Finally, the area of soft robotics and morphological computation seems to be rife with different *trading spaces* [63]. As we move from the traditional engineering framework with a central controller that commands a "dumb" mechanical structure toward delegating more functionality (control, perceptual, and computational) to the body, some convenient properties will be lost. In particular, the solutions may not be portable to other platforms anymore, as they will become dependent on the particular morphology and environment (a passive dynamic walker is the extreme case). The versatility of the solutions is likely to drop as well. To some extent, the morphology itself can be used to alleviate these difficulties—if it becomes adaptive. Online changes of morphology (such as changes of stiffness or shape) thus constitute another tough technological challenge (see [51] and project locomorph.eu). Completely new, distributed control algorithms that rely on self-organizing properties of complex bodies and local distributed control units will need to be developed (the tensegrity structure controlled by a spiking neural network [69] is a step in this direction).

The right mix and coupling of computational and morphological building blocks will always depend on the task at hand. Some subtasks may have a clear computational nature and may be simply best implemented on a microprocessor. Some, on the other hand, may greatly profit from exploiting the morphology; some may even disappear altogether if a different body design is employed. This article will hopefully help the practitioners to navigate in this complex landscape.

Acknowledgments

M.H. was supported by the Swiss National Science Foundation Fellowship PBZHP2-147259 and later by the Marie Curie Intra European Fellowship "iCub Body Schema" (625727) within the 7th European Community Framework Programme. M.H. also thanks Juan Pablo Carbajal for fruitful discussions and pointers to literature. V.C.M. was supported by a James Martin Research Fellowship at Oxford. Both authors thank our two reviewers of *Artificial Life* for insightful comments; the EUCogIII project (FP7-ICT 269981, coordinated by VCM) for making us talk to each other; and Rolf Pfeifer for making us think about these matters.

References

1. Albu-Schaffer, A., Eiberger, O., Grebenstein, M., Haddadin, S., Ott, C., Wimbock, T., Wolf, S., & Hirzinger, G. (2008). Soft robotics. *Robotics Automation Magazine, IEEE*, *15*(3), 20–30.

2. Autumn, K., Sitti, M., Liang, Y. A., Peattie, A. M., Hansen, W. R., Sponberg, S., Kenny, T. W., Fearing, R., Israelachvili, J. N., & Full, R. J. (2002). Evidence for van der Waals adhesion in gecko setae. *Proceedings of the National Academy of Sciences of the U.S.A.*, *99*(19), 12252–12256.

3. Bajcsy, R. (1988). Active perception. *Proceedings of the IEEE*, *76*(8), 966–1005.

4. Beer, R. D. (2003). The dynamics of active categorical perception in an evolved model agent. *Adaptive Behavior*, *11*(4), 209–243.
5. Blickhan, R., Seyfarth, A., Geyer, H., Grimmer, S., Wagner, H., & Guenther, M. (2007). Intelligence by mechanics. *Transactions of the Royal Society, London A*, *365*, 199–220.
6. Boyer, F., & Lebastard, V. (2012). Exploration of objects by an underwater robot with electric sense. In T. Prescott, N. Lepora, A. Mura, & P. Verschure (Eds.), *Biomimetic and biohybrid systems* (pp. 50–61). Berlin: Springer.
7. Boyer, F., Lebastard, V., Chevallereau, C., Mintchev, S., & Stefanini, C. (2015). Underwater navigation based on passive electric sense: New perspectives for underwater docking. *International Journal of Robotics Research*, *34*(9), 1228–1250.
8. Brown, E., Rodenberg, N., Amend, J., Mozeika, A., Steltz, E., Zakin, M. R., Lipson, H., & Jaeger, H. M. (2010). Universal robotic gripper based on the jamming of granular material. *Proceedings of the National Academy of Sciences of the U.S.A.*, *107*(44), 18809–18814.
9. Calvo-Garzón, P., & Gomila, A. (Eds.) (2008). *Handbook of cognitive science: An embodied approach*. Amsterdam: Elsevier.
10. Caluwaerts, K., & Schrauwen, B. (2011). The body as a reservoir: Locomotion and sensing with linear feedback. In H. Sumioka (Ed.), *2nd International Conference on Morphological Computation (ICMC2011)* (pp. 100–103).
11. Caluwaerts, K., D'Haene, M., Verstraeten, D., & Schrauwen, B. (2013). Locomotion without a brain: Physical reservoir computing in tensegrity structures. *Artificial Life*, *19*(1), 35–66.
12. Carbajal, J. P. (2012). *Harnessing nonlinearities: Behaviour generation from natural dynamics*. PhD thesis, University of Zurich.
13. Church, A. (1936). An unsolvable problem in elementary number theory. *The American Journal of Mathematics*, *58*, 345–363.
14. Collins, S., Ruina, A., Tedrake, R., & Wisse, M. (2005). Efficient bipedal robots based on passive dynamic walkers. *Science*, *307*, 1082–1085.
15. Collins, S. H., Wisse, M., & Ruina, A. (2001). A three-dimensional passive-dynamic walking robot with two legs and knees. *The International Journal of Robotics Research*, *20*(7), 607–615.
16. Crutchfield, J., Ditto, W., & Sinha, S. (2010). Introduction to focus issue. Intrinsic and designed computation: Information processing in dynamical systems—beyond the digital hegemony. *Chaos*, *20*(037101), 1–6.
17. Culha, U., Nurzaman, S. G., Clemens, F., & Iida, F. (2014). SVAS(3): Strain vector aided sensorization of soft structures. *Sensors*, *14*(7), 12748–12770.
18. Dambre, J., Verstraeten, D., Schrauwen, B., & Massar, S. (2012). *Information processing capacity of dynamical systems*. Nature Scientific Reports 2, 514.
19. Degrave, J., Caluwaerts, K., Dambre, J., & Wyffels, F. (2015). Developing an embodied gait on a compliant quadrupedal robot. In J. Zhang & A. Knoll (Eds.), *Intelligent robots and systems (IROS), 2015 IEEE/RSJ International Conference* (pp. 4486–4491). New York: IEEE.
20. Dodig-Crnkovic, G. (2011). Significance of models of computation, from Turing model to natural computation. *Minds and Machines*, *21*(2), 301–322.
21. Dodig-Crnkovic, G., & Müller, V. C. (2011). A dialogue concerning two world systems: Info-computational vs. mechanistic. In G. Dodig-Crnkovic & M. Burgin (Eds.), *Information and computation: Essays on scientific and philosophical understanding of foundations of information and computation* (pp. 149–184). Boston: World Scientific.
22. Fernando, C., & Sojakka, S. (2003). Pattern recognition in a bucket. In W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, & J. Kim (Eds.), *Advances in artificial life* (pp. 588–597). Heidelberg: Springer.
23. Floreano, D., Pericet-Camara, R., Viollet, S., Ruffier, F., Brückner, A., Leitel, R., Buss, W., Menouni, M., Expert, F., Juston, R. Dobrzynski, M. K., L'Eplattenier, G., Reckenwald, F., Mallot, H. A., & Franceschini, N. (2013). Miniature curved artificial compound eyes. *Proceedings of the National Academy of Sciences of the U.S.A.*, *110*(23), 9267–9272.
24. Floridi, L. (2004). Open problems in the philosophy of information. *Metaphilosophy*, *35*(4), 554–582.

25. Franceschini, N., Pichon, J., & Blanes, C. (1992). From insect vision to robot vision. *Transactions of the Royal Society, London B*, *337*, 283–294.

26. Füchslin, R., Dzyakanchuk, A., Flumini, D., Hauser, H., Hunt, K., Luchsinger, R., Reller, B., Scheidegger, S., & Walker, R. (2013). Morphological computation and morphological control: Steps towards a formal theory and applications. *Artificial Life*, *19*(1), 9–34.

27. Hauser, H., Ijspeert, A., Füchslin, R., Pfeifer, R., & Maass, W. (2011). Towards a theoretical foundation for morphological computation with compliant bodies. *Biological Cybernetics*, *105*(5), 355–370.

28. Hauser, H., Ijspeert, A., Füchslin, R., Pfeifer, R., & Maass, W. (2012). The role of feedback in morphological computation with compliant bodies. *Biological Cybernetics*, *106*(10), 595–613.

29. Hermans, M., Schrauwen, B., Bienstman, P., & Dambre, J. (2014). Automated design of complex dynamic systems. *PLoS ONE*, *9*(1), e86696.

30. Hoffmann, M., & Müller, V. C. (forthcoming). Simple or complex bodies? Trade-offs in exploiting body morphology for control. In G. Dodig-Crnkovic & R. Giovagnoli (Eds.), *Representation of reality: Humans, animals and machines*. Berlin: Springer. Available at http://www.springer.com/qp/book/9783319437828.

31. Hoffmann, M., & Pfeifer, R. (2011). The implications of embodiment for behavior and cognition: Animal and robotic case studies. In W. Tschacher & C. Bergomi (Eds.), *The implications of embodiment: Cognition and communication* (pp. 31–58). Exeter: Imprint Academic.

32. Horsman, C., Stepney, S., Wagner, R., & Kendon, V. (2014). When does a physical system compute? *Proceedings of the Royal Society A*, *470*(2169), 20140182, 1–25.

33. Iida, F., Gómez, G., & Pfeifer, R. (2005). Exploiting body dynamics for controlling a running quadruped robot. In B. Hannaford (Ed.), *Proceedings of the 12th International Conference on Advanced Robotics (ICAR05)* (pp. 229–235). New York: IEEE.

34. Ikemoto, S., Nishigori, Y., & Hosoda, K. (2012). Advantages of flexible musculoskeletal robot structure in sensory acquisition. *Artificial Life and Robotics*, *17*(1), 63–69.

35. Ilievski, F., Mazzeo, A. D., Shepherd, R. F., Chen, X., & Whitesides, G. M. (2011). Soft robotics for chemists. *Angewandte Chemie*, *123*(8), 1930–1935.

36. Jäger, H. (2002). *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach*. GMD Report 159, German National Research Center for Technology. Available at minds.jacobs-university.de/sites/default/files/uploads/papers/ESNTutorialRev.pdf (accessed May 2016).

37. Jäger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, *304*(5667), 78–80.

38. Johnson, C., Philippides, A., & Husbands, P. (2016). Active shape discrimination with compliant bodies as reservoir computers. *Artificial Life*, *22*(2), 241–268.

39. Kim, S., Spenko, M., Trujillo, S., Heyneman, B., Mattoli, V., & Cutkosky, M. R. (2007). Whole body adhesion: Hierarchical, directional and distributed control of adhesive forces for a climbing robot. In P. Fiorini (Ed.), *Robotics and Automation, 2007 IEEE International Conference* (pp. 1268–1273). New York: IEEE Press.

40. Kleene, S. C. (1952). *Introduction to metamathematics*. Amsterdam: North Holland.

41. Koch, C. (1999). *Biophysics of computation: Information processing in single neurons*. New York: Oxford University Press.

42. Koditschek, D. E., Full, R. J., & Buehler, M. (2004). Mechanical aspects of legged locomotion control. *Arthropod structure and development*, *33*, 251–272.

43. Kurowski, S., & von Stryk, O. (2015). A systematic approach to the design of embodiment with application to bio-inspired compliant legged robots. In J. Zhang & A. Knoll (Eds.), *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference* (pp. 3771–3778). New York: IEEE.

44. Larger, L., Soriano, M., Brunner, D., Appeltant, L., Gutierrez, J., Pesquera, L., Mirasso, C., & Fischer, I. (2012). Photonic information processing beyond Turing: An optoelectronic implementation of reservoir computing. *Optics Express*, *20*(3), 3241–3249.

45. Lind, D. A. (1995). *An introduction to symbolic dynamics and coding*. Cambridge, UK: Cambridge University Press.

46. Lipson, H. (2013). Challenges and opportunities for design, simulation, and fabrication of soft robots. *Soft Robotics*, *1*(1), 21–27.

47. Lucosevicus, M., & Jaeger, H. (2009). Survey: Reservoir computing approaches to current neural network training. *Computer Science Review*, *3*(3), 127–149.

48. Lungarella, M., & Sporns, O. (2006). Mapping information flow in sensorimotor networks. *PLoS Computational Biology*, *2*, 1301–1312.

49. Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, *14*(11), 2531–2560.

50. Mammano, F., & Nobili, R. (1993). Biophysics of the cochlea: Linear approximation. *The Journal of the Acoustical Society of America*, *93*, 3320.

51. McEvoy, M., & Correll, N. (2015). Materials that couple sensing, actuation, computation, and communication. *Science*, *347*(6228), 1261689.

52. McGeer, T. (1990). Passive dynamic walking. *The International Journal of Robotics Research*, *9*(2), 62–82.

53. Moore, A., Ober-Blöbaum, S., & Marsden, J. E. (2012). Trajectory design combining invariant manifolds with discrete mechanics and optimal control. *Journal of Guidance, Control, and Dynamics*, *35*(5), 1507–1525.

54. Müller, V. C. (2009). Symbol grounding in computational systems: A paradox of intentions. *Minds and Machines*, *19*(4), 529–541.

55. Müller, V. C. (2011). On the possibilities of hypercomputing supertasks. *Minds and Machines*, *21*(1), 83–96.

56. Müller, V. C. (2014). Pancomputationalism: Theory or metaphor? In R. Hagengruber & U. Riss (Eds.), *Philosophy, computing and information science* (pp. 213–221). London: Pickering & Chattoo.

57. Nakajima, K., Hauser, H., Kang, R., Guglielmino, E., Caldwell, D. G., & Pfeifer, R. (2013). A soft body as a reservoir: Case studies in a dynamic model of octopus-inspired soft robotic arm. *Frontiers in Computational Neuroscience*, *7*(91). doi:10.3389/fncom.2013.00091.

58. Nakajima, K., Hauser, H., Li, T., & Pfeifer, R. (2015). Information processing via physical soft body. *Nature Scientific Reports*, *5*(10487). doi:10.1038/srep10487.

59. Nurzaman, S., Yu, X., Kim, Y., & Iida, F. (2014). Guided self-organization in a dynamic embodied system based on attractor selection mechanism. *Entropy*, *16*(5), 2592–2610.

60. Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., & Massar, S. (2012). Optoelectronic reservoir computing. *Nature Scientific Reports*, *2*(287). doi:10.1038/srep00287.

61. Pfeifer, R., & Bongard, J. C. (2007). *How the body shapes the way we think: A new view of intelligence*. Cambridge, MA: MIT Press.

62. Pfeifer, R., Lungarella, M., & Iida, F. (2012). The challenges ahead for bio-inspired 'soft' robotics. *Communications of the ACM*, *55*(11), 76–87.

63. Pfeifer, R., Marques, H. G., & Iida, F. (2013). Soft robotics: The next generation of intelligent machines. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 5–11).

64. Pearson, M. J., Mitchinson, B., Sullivan, J. C., Pipe, A. G., & Prescott, T. J. (2011). Biomimetic vibrissal sensing for robots. *Philosophical Transactions of the Royal Society B*, *366*(1581), 3085–3096.

65. Piccinini, G. (2003). *Computations and computers in the sciences of mind and brain*. PhD thesis, University of Pittsburgh. Available at http://etd.library.pitt.edu/ETD/available/etd-08132003-155121/.

66. Piccinini, G. (2011). The physical Church-Turing thesis: Modest or bold? *British Journal for the Philosophy of Science*, *62*(4), 733–769.

67. Prescott, T. J., Diamond, M. E., & Wing, A. M. (2011). Active touch sensing. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, *366*(1581), 2989–2995.

68. Prescott, T. J., Pearson, M. J., Mitchinson, B., Sullivan, J. C. W., & Pipe, A. G. (2009). Whisking with robots from rat vibrissae to biomimetic technology for active touch. *IEEE Robotics and Automation Magazine*, *16*(3), 42–50.

69. Rieffel, J. A., Valero-Cuevas, F. J., & Lipson, H. (2010). Morphological communication: Exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of the Royal Society Interface*, *7*(45), 613–621.

70. Rückert, E., & Neumann, G. (2013). Stochastic optimal control methods for investigating the power of morphological computation. *Artificial Life*, *19*, 115–131.

71. Saranli, U., Buehler, M., & Koditschek, D. (2001). RHex: A simple and highly mobile hexapod robot. *International Journal of Robotics Research*, *20*, 616–631.

72. Smerieri, A., Duport, F., Paquot, Y., Schrauwen, B., Haelterman, M., & Massar, S. (2012). Analog readout for optical reservoir computers. Available at http://arxiv.org/abs/1209.3129 (accessed May 2016).

73. Song, Y. M., Xie, Y., Malyarchuk, V., Xiao, J., Jung, I., Choi, K.-J., Liu, Z., Park, H., Lu, C., Kim, R.-H., Li, R., Crozier, K. B., Huang, Y., & Rogers, J. A. (2013). Digital cameras with designs inspired by the arthropod eye. *Nature*, *497*(4774), 95–99.

74. Sornkarn, N., Howard, M., & Nanayakkara, T. (2014). Internal impedance control helps information gain in embodied perception. In Y. Liu (Ed.), *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 6685–6690). New York: IEEE.

75. Swapnil, P., DeForest, M., Edward, J. B., & Matthew, A. R. (2015). Effects of sensilla morphology on mechanosensory sensitivity in the crayfish. *Bioinspiration & Biomimetics*, *10*(3), 036006.

76. Taga, G. (1995). A model of the neuro-musculo-skeletal system for human locomotion I. Emergence of basic gait. *Biological Cybernetics*, *73*, 97–111.

77. Taga, G. (1995). A model of the neuro-musculo-skeletal system for human locomotion II. Real-time adaptability under various constraints. *Biological Cybernetics*, *73*, 113–121.

78. Trimmer, B. (2013). A journal of soft robotics: Why now? *Soft Robotics*, *1*(1), 1–4.

79. Trivedi, D., Rahn, C., Kier, W., & Walker, I. (2008). Soft robotics: Biological inspiration, state of the art, and future research. *Applied Bionics and Biomechanics*, *5*(3), 99–117.

80. Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, *42*, 230–256.

81. Valero-Cuevas, F. J., Yi, J.-W., Brown, D., McNamara, R. V., Paul, C., & Lipson, H. (2007). The tendon network of the fingers performs anatomical computation at a macroscopic scale. *IEEE Transactions on Biomedical Engineering*, *54*(6), 1161–1166.

82. Varela, F. J., Thompson, E., & Rosch, E. (1991). *The embodied mind: Cognitive science and human experience*. Cambridge, MA: MIT Press.

83. Wiesner, K., & Crutchfield, J. P. (2008). Computation in finitary stochastic and quantum processes. 28.04.2008. Available at http://arxiv.org/abs/quant-ph/0608206 (accessed May 2016).

84. Zenil, H. (2013). A behavioural foundation for natural computing and a programmability test. In G. Dodig-Crnkovic & R. Giovagnoli (Eds.), *Computing nature: Turing centenary perspective* (pp. 87–113). Berlin, Heidelberg: Springer.