Morphological Computation and Morphological Control: Steps Toward a Formal Theory and Applications

Rudolf M. Füchslin*,**
Zurich University of Applied Sciences

Andrej Dzyakanchuk†
Bern University of Applied Sciences

Dandolo Flumini‡
Zurich University of Applied Sciences

Helmut Hauser§
University of Zurich

Kenneth J. Hunt¶
Bern University of Applied Sciences

Rolf H. Luchsinger#
Empa—Center for Synergetic Structures

Benedikt Reller††
European Center for Living Technology

Stephan Scheidegger**
Zurich University of Applied Sciences

Richard Walker‡‡
Blue Brain Project

Abstract Morphological computation can be loosely defined as the exploitation of the shape, material properties, and physical dynamics of a physical system to improve the efficiency of a computation. Morphological control is the application of morphological computing to a control task. In its theoretical part, this article sharpens and extends these definitions by suggesting new formalized definitions and identifying areas in which the definitions we propose are still inadequate. We go on to describe three ongoing studies, in which we are applying morphological control to problems in medicine and in chemistry. The first involves an inflatable support system for patients with impaired movement, and is based on macroscopic physics and concepts already tested in robotics. The two other case studies (self-assembly of chemical microreactors; models of induced cell repair in radio-oncology) describe processes and devices on the micrometer scale, in which the emergent dynamics of the underlying physical system (e.g., phase transitions) are dominated by stochastic processes such as diffusion.

Keywords
Morphological computation, embodiment, analogue computation, systems medicine, prosthetics

A version of this paper with color figures is available online at http://dx.doi.org/10.1162/artl_a_00079. Subscription required.

* Contact author.
** Center for Applied Mathematics and Physics, School of Engineering, Zurich University of Applied Sciences, Tecnikumstr. 9, CH-8400, Winterthur, Switzerland. E-mail: rudolf.fuechslin@zhaw.ch (R.M.F.); stephan.scheidegger@zhaw.ch (S.S.)
† Institute for Rehabilitation and Performance Technology, Division of Mechanical Engineering, Department of Engineering and Information Technology, Bern University of Applied Sciences, CH-3400 Burgdorf, Switzerland. E-mail: andrei.dzyakanchuk@bfh.ch
‡ School of Engineering, Zurich University of Applied Sciences, Lagerstrasse 41 8004 Zürich, Switzerland. E-mail: dandolo.flumini@zhaw.ch
§ Artificial Intelligence Laboratory, Department of Informatics, University of Zurich, Andreasstrasse 15, CH-8050 Zurich, Switzerland. E-mail: hhauser@ifi.uzh.ch
¶ Institute for Rehabilitation and Performance Technology, Division of Mechanical Engineering, Department of Engineering and Information Technology, Bern University of Applied Sciences, CH-3400 Burgdorf, Switzerland. E-mail: kenneth.hunt@bfh.ch
# Empa—Center for Synergetic Structures, Ueberlandstrasse 129, CH-8600 Dübendorf, Switzerland. E-mail: Rolf.Luchsinger@empa.ch
†† European Centre for Living Technology, Ca' Minich, S. Marco 2940, 30124 Venezia, Italy. E-mail: bene.reller@gmail.com
‡‡ Blue Brain Project, EPFL, QIJ 3 115.6 CH-1015 Lausanne, Switzerland. E-mail: richard.walker@epfl.ch

1 Introduction

Some of the most important current applications of morphological computing are in robotics, a field in which it is widely accepted that the intrinsic dynamics of a physical system (the robot body) need not be regarded only as a potential source of perturbations, but can be exploited to facilitate the control of the system [28]. Control implies computation. The use of the dynamics of the robot body to facilitate control can thus be interpreted as an *outsourcing* of computational tasks to the body. In this interpretation, the body is performing *morphological computation* [25].

In robotics, the concept of morphological computing has been around for some time, though the term itself is relatively recent (see [17] and, for a broader presentation, [27]). Today, however, it is clear that the concept of morphological computing can be applied not just to systems such as robots, whose dynamics are governed by classical mechanics, but also to systems, such as self-assembling and/or chemical systems, that are better described in terms of statistical physics [29].

In 2007, a workshop at the first International Conference on Morphological Computing in Venice, Italy, led by Norman Packard, informally defined morphological computing as "any process that (a) serves for a computational purpose, (b) has clearly assignable input and output states and (c) is programmable, where 'programmable' is understood in the broad sense that a programmer can vary the behavior of the system by varying a set of parameters." This definition implies that conventional digital computing systems are instances of morphological computing. However, many researchers require that at least part of the computation should rely in a significant way on the physical dynamics of the system implementing the computation, which should be entirely defined by the "program." By this definition, the execution of a conventional computer program that completely specifies the transformation of an input to an output cannot be classed as morphological computation. The formalization we propose in Section 2.2 casts new light on this issue.

The rest of the article will be structured as follows. We begin by discussing conceptual aspects of morphological computation and morphological control. We then examine how these concepts apply to robotics and to more general computational applications. On this basis, we suggest a more formalized definition. Our definition is a suggestion and reflects our point of view; in what follows "morphological computation" should be interpreted as "the view of morphological computation held by the authors." The article concludes with three case studies, two of which go beyond robotics and the underlying classical mechanics to consider systems whose dynamics are governed by statistical mechanics.

2 Morphological Computation and Control

To begin, we would like to clarify a number of issues that often arise in discussions.

1. The concept of morphological computation, as we understand it, does not imply any notion of "hypercomputation." All the computational and/or control problems we discuss in our case studies are computable by a conventional Turing machine. Unlike the Turing machine, morphological computation does not define a class of computational problems. Rather it provides a method for solving such problems: In Marr's terminology, the concept belongs not to the "computational level" (the level of the Turing machine) but to the "implementation level" [22].
2. In our work we use "morphology" to mean the combination of shape and material properties—elasticity, friction coefficients and so on; by this definition, morphology includes material as well as geometrical properties.
3. In what follows, we distinguish between computation and control. In our terminology, "morphological control" means "control achieved via morphological computation."

4. The concepts of morphological computation and control overlap to some extent with control theory. However, in control theory, the crucial notion is that of the *reference*, which provides an exact specification of the way system parameters should behave. In morphological control, as we will see, no reference is explicitly required.

5. Morphological computing also overlaps to a large extent with what has previously been called *natural computing* [15], the attempt to implement conventional computation processes using non-electronic means, for example, DNA computation or membrane computation. However, the two fields have different emphases. Much of the literature on natural computing is dedicated to demonstrations that a specific system of natural computing is universal, in the sense that it can compute all problems that can be computed by a Turing machine. The literature on morphological computation attaches much less importance to this concept. In most cases, the goal of morphological computing is to control specific devices in potentially broad but specified contexts, which may or may not allow the representation of a particular class of mathematical problem (e.g., a Hamiltonian graph problem). A second difference in emphasis is in the role of electronics. Most of the natural computing techniques reported in the literature do not use electronics at all. Many morphological computing and control systems continue to use some kind of conventional control unit.

6. In the view we present here, the main goal of morphological control is to perform control tasks in a more efficient manner than would be possible with conventional computing technology, where efficiency is understood in in terms of time, memory, power, space, and other costly resources. This is a tradeoff: The gain in efficiency is achieved by abandoning the strict separation of hardware and software, one of the fundamental characteristics of conventional control. Morphological control may also provide looser control than conventional computing technologies.

2.1 From Robotics to General Models of Morphological Computation

Conventionally, the control of a robot follows the scheme given in Figure 1A. Here, a robot gets information from the environment via a sensory system. These signals are then transformed into a binary representation, which is processed by a conventional computer. The result of this computation is used to determine the action of the robot. In most cases, the conventional computation is based on a kinematic model of the robot and uses classical mechanics to evaluate relevant aspects of the robot dynamics. The kinematic model requires knowledge of the state of the robot. Given that the free parameters of the robot (angles of joints, velocities, etc.) can only be determined with finite accuracy and with a finite sampling rate, the control unit has to deal with *physical noise*. Thus, to keep the computation as simple and reliable as possible, control systems are designed to minimize the undesirable influence of the robot's morphology (seen as a source of physical noise). One strategy is to keep the number of degrees of freedom as low as possible. From a physical point of view, this is one of the reasons why most robots are heavy and stiff. Under these conditions, a limited number of system parameters can give an account of the state of the robot (e.g., its posture and velocities) that is sufficiently complete and accurate to allow effective control.

Morphological control pursues an alternative strategy, described in Figure 1B. Now the control system is designed not to dampen the robot's natural dynamics, but to exploit them. As we will discuss, it is no longer necessary to represent the complete state of the robot in the control unit. As a consequence, robots with morphological control can use soft materials with many internal degrees of freedom. Using soft and lightweight materials offers many benefits. In health care, for instance, they usually raise fewer safety concerns than harder, heavier materials.

Although biological systems usually involve more sophisticated control schemes, the way the physical dynamics of a system can be used for control can best be explained by a simple example (see Figure 2). Assume a (sub)system determined by two parameters. These parameters are subject

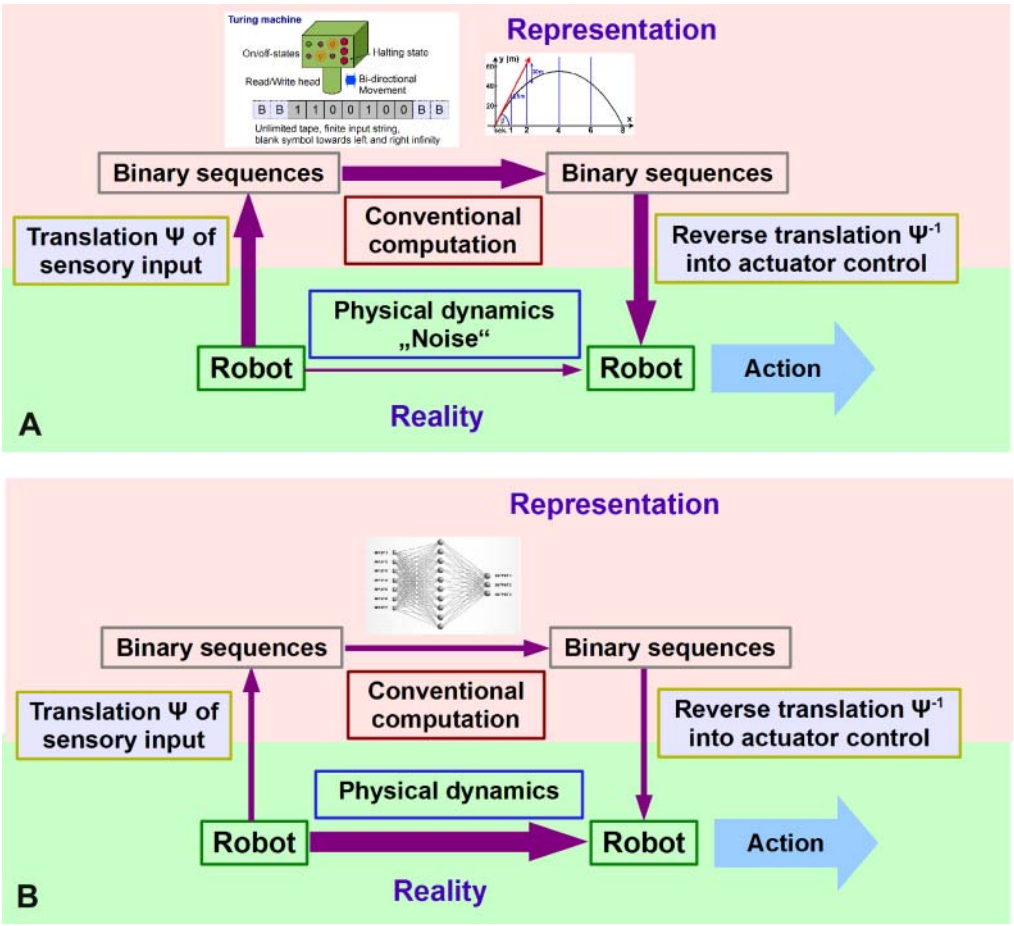

Figure 1. (A) Conventional control scheme, (B) control scheme using morphological control. The lower panel refers to the physical realization of the systems under consideration. The upper panel refers to the digital representation of the same system. The width of the arrows represents the relative importance of different flows of information. In conventional control, the ideal situation is achieved when all relevant information is passed through the digital control, that is, when the control unit (implemented by a conventional digital computer) has complete information about the state of the system, and the system performs its commands with maximal precision. In reality, this ideal situation is never reached, but good engineering aims to minimize the non-controlled aspects of the system. Thus, the lower horizontal arrow in (A) is as thin as possible. (B) Illustrates morphological control. The amount of information passed through the digital control is as small as possible and the conventional computation is as simple as possible (e.g., a feed-forward neural net). As far as possible, the computation exploits the physical dynamics of the system.

to dynamics that create two limit cycles, each with its own basin of attraction. (Note that the dynamical systems under consideration are usually dissipative, meaning they take up energy.) This means they can display limit cycles, despite the presence of friction. To move between the two basins of attraction, the control unit does not need to regulate every detail of the system's trajectory. All it has to do is choose some initial (x_0, y_0) lying in the right basin of attraction. Small perturbations are corrected naturally by the tendency of the system to choose one of the two limit cycles. In other words, the limit cycles significantly simplify the computational task for the control unit. In classical control, the controller needs to generate every detail of a trajectory and to handle perturbations. In morphological control, all it has to do is choose a basin of attraction. Of course, there is a price to pay: While conventional control allows a large variety of different trajectories (and gives complete control over transients), the simple system in the left panel of Figure 2 is only able to produce two loosely defined oscillatory movements.

In reality, the concept of morphological control means much more than simply exploiting the self-stabilizing properties of dynamical systems (standard practice in good engineering). Control systems can also change the properties of a system, altering the structure of its attractor landscape. Such changes can result in a modification of the limit cycle, the basins of attraction, and even the number and structure of attractors (see Figure 2, right side). In more general terms, we can construct the system in such a way that a small number of internal system parameters (such as internal tension) can generate a rich variety of movement patterns. The control unit basically performs two tasks: First, it shapes the attractor landscape of the dynamical system it controls; second, it initiates jumps from one basin of attraction to another. In this way, the calibration and stabilization of a movement pattern are outsourced to the dynamics of the system.

Moving from robotics to physiology, we can use similar concepts to interpret the behavior of natural systems. For example, consider the way one changes one's posture when carrying a heavy load. Though we have no strict proof, we interpret the change as a reshaping of the body's attractor landscape. The human body is a dynamical system that is optimized for walking under normal conditions. A heavy load changes the properties of the system. To readjust we alter the positions of some of our joints and change the tension on some of our muscles. This seems to be a form of morphological control.

Another example of the interplay between physiological dynamics, control by the brain, and technology is downhill skiing. Skiers continuously adjust their posture to the terrain in order to gain stability against perturbations (without losing the ability to steer). At the same time, the choice of the mechanical properties of skis and boots influences the dynamical behavior of the "skier," considered as an integrated system [2, 24]. We hypothesize that these changes represent forms of morphological control. For example, certain types of skis (carving skis) are generally regarded as easier to use than the skis used by athletes. In other words, they reduce the complexity of the control problem posed by downhill skiing. However, they also offer less precise control. This explains why racers choose different, harder-to-use models.

We hypothesize that, in the two cases we have just considered (walking, downhill skiing), the brain plays an important role not just in choosing between preexisting basins of attraction, but in reshaping the attractor landscape (e.g., when a skier needs to change direction or vary speed). It does not need to control all of the body's free parameters to achieve this. Rather, it learns to control a small number of parameters governing posture and internal tension, thereby altering the overall dynamics of

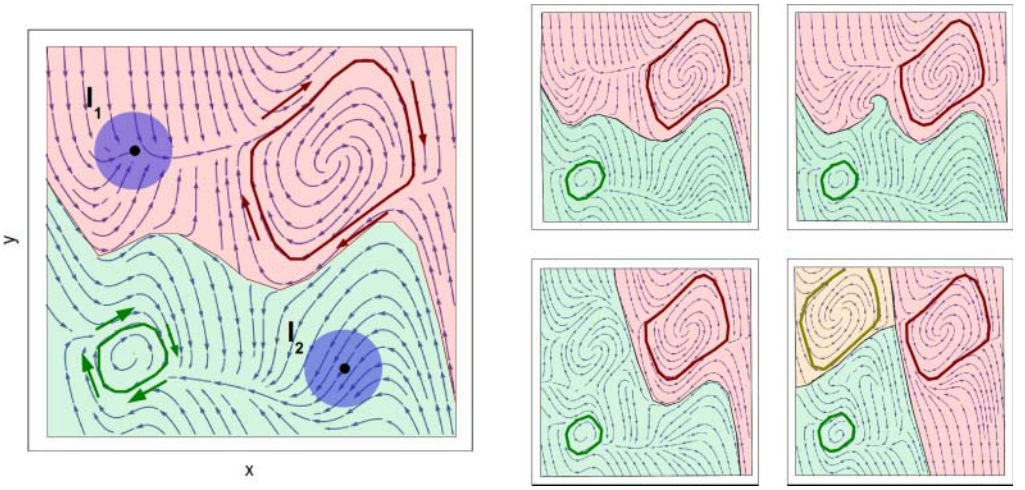

Figure 2. Left: Schematic example of self-regulation of a dynamical system. Two system parameters x and y are subject to a dynamics that leads to limit cycles, sketched together with their respective basins of attraction. I_1 and I_2 denote initial conditions: An observer is not able to prepare a system in an exactly specified initial state, but may guarantee to choose a point within a certain finite part of the parameter space. Right: Dynamic attractor landscape. Upon change of a system parameter, the basins of attraction, the shape of the limit cycles, and even the number of attractors is altered.

the body and drastically reducing the complexity of the control task. The hypothesis that the brain behaves in this way is supported by psychological findings showing that humans organize complex movements into subtasks, represented in memory by morphologies [35].

A further layer of sophistication can be added by reading out certain system parameters and processing the results. For example, it is well established that the dynamics of artificial neural networks can be used to perform complex computations [14, 20]. Hauser et al. have recently demonstrated [10] that similar results hold for compliant bodies: a (sufficiently complex) mass-spring system coupled to a feed-forward neural network can, in principle, approximate a large class of input-output relations (time-invariant filters with fading memory) with arbitrary precision.

In summary, we can establish a hierarchy of different forms of morphological computation:

1. Switching between basins of attraction under the control of a simple control signal.
2. Control of the attractor landscape by a simple control system.
3. Postprocessing of the state of a dynamical system to obtain a computational result.

This last level is particularly interesting when the system in question is an animal or a robot. In this case, the information gained by postprocessing can be translated into control signals and fed back into the system as input signals.

One may also extend this hierarchy to take account of learning: changes in the program of the conventional controller. To date, however, we have yet to achieve a theory of learning that combines abstracted and physical aspects of control.

2.2 Morphological Computation in General: Toward a Definition

So far we have based our discussion on an informal definition of morphological computation. In this section we take a more formal approach, which, while not directly applicable to all aspects of our case studies, establishes a connection to theoretical computer science.

As we saw earlier, the Venice definition states that a natural system is a morphological computing system if and only if

1. it converts reproducible input into a reproducible output;
2. it is programmable in the sense that the map between input and output is parameterized in such a way that a wide variety of outputs can be produced;
3. it has a sort of teleological embedding.

Requirement 1 has three important consequences. First, the notion of a "conversion" implies that the system under consideration is a dynamical system. Second, reproducibility requires that the system should be deterministic in the strong sense. Third, the same condition requires that it should be possible to tune the system so that two indistinguishable input states lead to the same output state. Note that "indistinguishable" does not mean "identical." In the example of a dynamical system with different basins of attraction, it is sufficient that the observer should be able to start the system in a given basin of attraction. This condition rules out continuous chaotic systems, whose behaviors can be completely different for arbitrarily similar inputs.

Requirement 2 requires that a computational system should be programmable at least to some extent. While equivalence to a universal Turing machine is not required, the system should be adaptable. Take as an example a passive dynamic walker [23]. The dynamics of the system should spontaneously compensate for small perturbations. However, programmability implies that it should be possible to adapt the system to different types of terrain (perhaps by adjusting spring constants, friction coefficients, etc.). Note that in this case, programmability does not require a formal programming language. The "program" may be a set of parameters.

The third requirement protects the concept of morphological computation from the trivial observation that any dynamical system is a "simulation" of itself and can thus be regarded as computing its own defining equations.

In order to develop a theory of morphological computation, formalized definitions are a necessity. As will be discussed, the definitions we suggest do not yet capture the whole range of processes one might consider as morphological computations. Part of the motivation for presenting our definition is that we claim to have worked out what is lacking to get a more comprehensive definition. Furthermore, our definition does not require the system to be discrete (as a conventional computer is).

2.2.1 Formalizing Morphological Computation

Our first task is to elaborate a suitable way to assign functions to dynamical systems. In other words, we have to give them a denotational semantics. A natural approach to this problem is to assign each system to the (partial) function that maps any initial state x to some state y, where y is the fixed point one reaches starting from x. However, we want to be more liberal in what we will allow as an output value, not restricting ourselves to fixed points. As output values, we will therefore admit forward invariant state sets that we call terminations.

Definition: A dynamical system Γ is a triple $\Gamma = (S, M, f)$ consisting of a set of states S, a partially ordered monoid $(M, <)$, and a transition function $f: S \times M \to S$ such that:

- $f(s, x + y) = f(f(s, x), y)$.
- The set M is linearly ordered by $<$.
- In order to guarantee that the addition of times in the first requirement is compatible with the monoid, we have to require that $(p, t \in M) \wedge (p < t) \Rightarrow \exists q \in M(p + q = t)$.

The set M is nothing else than the set of time values. In what follows, it will always hold that $M = \mathbb{R}$.

Definition: Given a dynamical system $\Gamma = (S, M, f)$, the *forward orbit* $O^+_\Gamma(x)$ of $x \in S$ is defined as the set of states $O^+_\Gamma(x) = \{p | p \in S, \exists t > 0\ (f(t, x) = p)\}$.

Definition: Given a dynamical system $\Gamma = (S, M, f)$, let T be a subset of S.

- T is called *forward closed* if $a \in T \Rightarrow O^+_\Gamma(a) \subset T$.
- T is called *forward connected* if $a, b \in T \Rightarrow a \in O^+_\Gamma(b)$. Note that this definition is symmetric, that is, it implicitly holds that $b \in O^+_\Gamma(a)$.
- T is called a *termination* if it is both forward closed and forward connected.

The set of all terminations of Γ will be denoted by $\Theta(\Gamma)$.

Accepting terminations as output values makes it possible to have a sequences of states as output—for example, in robotics, the sequence of states representing a movement.

Definition: The *functional dynamics* of a dynamical system $\Gamma = (S, M, f)$ is the (partial) function d_Γ that sends, if possible, an (initial) state x to a termination T if $O^+_\Gamma(x)$ has a nontrivial intersection with T.

Note that if the system Γ starting with initial condition x after a transient eventually stabilizes in an oscillatory movement given by the termination T, then we have $d_\Gamma(x) = T$.

Now, we introduce the concept of a programmable dynamical system.

Definition: A *programmable dynamical system* is a triple $\Pi = (S, M, (f_j)_{j \in J})$, where for each $j \in J$ the triple (S, T, f_j) is a dynamical system. An element $j \in J$ is a *program* for P.

The definitions given up to this point help us to discuss morphological control. They (and the concept of an observer to be detailed below) will be used in the discussion of the case studies. In what follows, we extend our formalization to treat morphological computations that map natural numbers onto other natural numbers, establishing contact with the theory of conventional computation, and showing how the concept of a programmable dynamical system can represent the first step toward a theory of morphological computation oriented toward computation in general. A future article will examine this possibility in greater depth.

First of all, we introduce the concept of an observer and specify what we mean when we say that a specific observer can use a system to compute a given function from the natural numbers to the natural numbers. The interpretation of such natural numbers as reals representing positions, angles, and so on is regarded as unproblematic.

We observe that there are three possible kinds of interaction between an observer and a computer: programming, setting input, and reading output. To formalize the notion of an observer of a given programmable dynamical system $\Pi = (S, M, (f_j)_{j \in J})$, we define a triple (I_Π, R_Π, p). In this setting, I_Π are input states (a subset of S), and R_Π is a binary relation defined on S with a value of 0 when two states cannot be distinguished and 1 otherwise. We impose the requirement that, for each pair $x, y \in I_\Pi$, we have $R_\Pi(x, y) = 0$. This allows us to handle the problem that the initial states of a continuous physical system can only be prepared with finite precision. Finally, the partial function p maps a number-theoretic function φ to the set J, the index set of the transition functions. (Note that each element j of J corresponds to a parameter setting.) The partial function p models the ability of the observer to set up the parameters of the system such that the system computes the function φ (see below). The situation is visualized in Figure 2. The set $I_\Pi = \{i_1, i_2\}$ consists of representatives (e.g., the centers, black dots) from each of the surrounding shaded regions. When preparing the system in state i_n the observer chooses a state in the surround of i_n. The condition that any two elements of I_Π can be distinguished ensures that the observer is able to check the input for correctness.

Before we can say under which conditions an observer is able to compute a given number-theoretic function, we have to connect the functional dynamics of a system with number-theoretic functions. We call this link a computational interpretation.

Definition: A *computational interpretation* Υ_Γ of a dynamical system $\Gamma = (S, M, f)$ is a pair (C_Γ, D_Γ) that consists of a coding function $C_\Gamma : \mathbb{N} \to S$ that maps natural numbers to initial states of Γ and a decoding function $D_\Gamma : \Theta(\Gamma) \to \mathbb{N}$ that maps terminations of Γ to the natural numbers.

Now, we can define what it means when we say that an observer can use a dynamical system to compute a partial number-theoretic function φ.

Definition: Given a programmable dynamical system $\Pi = (S, M, (f_j)_{j \in J})$ and a computational interpretation $\Upsilon_\Pi = (C_\Pi, D_\Pi)$, the observer (I_Π, R_Π, p) can use Π to compute the partial function $\varphi : \mathbb{N} \to \mathbb{N}$ relative to the computational interpretation Υ_Π if for each natural number n:

- The function p is defined on φ.
  Remark: The purpose of this condition is to make sure that there is a program to implement φ.
- $C_\Pi(n)$ is an element of I_Π.
  Remark: The purpose of this condition is to make sure that the observer is capable of setting up the system so that it computes the function for all desired values.
- For any natural number k, if $d_{\Pi, p(\varphi)}(C_\Pi(n))$ is different from $d_{\Pi, p(\varphi)}(C_\Pi(k))$, there exist two states $q \in d_{\Pi, p(\varphi)}(C_\Pi(n))$ and $s \in d_{\Pi, p(\varphi)}(C_\Pi(k))$, respectively, such that $R_\Pi(q, s) = 1$.
  Remark: This condition allows an observer to use a system to compute two different values of the function φ only in the case that the observer is capable of distinguishing the two related outputs.

- $\varphi(n) = D_\Pi\,(d_{\Pi,p(\varphi)}\,(C_\Pi(n)))$.
  Remark: This last item connects the functional dynamics of Π_i with the function φ via the interpretation (C_Π, D_Π).

The computation of φ using the dynamics of Π is called a *morphological computation*. Basically, the definition of morphological computation states that the diagram

$$\begin{array}{ccc} S & \overset{d_{\Pi,j}}{\rightarrow} & \Theta(\Pi_i) \\ \uparrow C_\Pi & & \downarrow D_\Pi \\ \mathbb{N} & \overset{\varphi}{\rightarrow} & \mathbb{N} \end{array} \tag{1}$$

is commutative and that all information processing required is accessible to the observer. Note that the coding and the decoding function do not depend on the program j, whereas the functional dynamics is determined by the parameter setting.

It is crucial to understand the relationship between the function φ, a conventional program, the functional dynamics $d_{\Pi,p(\varphi)}$, and the program $p(\varphi)$ determining the functional dynamics given the dynamical system Π.

The function φ is a mathematical object that does not describe how its output $\varphi(n)$ is generated from an input n. A conventional program is a string over a certain alphabet and describes an algorithm that makes it possible to compute $\varphi(n)$ on a standardized device such as a universal Turing machine. This string, together with the description of a Turing machine, contains all the information necessary for the evaluation of $\varphi(n)$. The program contains no reference to the physical properties of the device that actually performs the computation. In the case of the functional dynamics of a programmable dynamical system, the program consists of a set of parameter values. This set, together with the dynamical system and the computational interpretation (C_Π, D_Π), makes it possible to compute $\varphi(n)$. Whereas in the case of a conventional computation an abstract description is sufficient, the morphological computation requires an actual physical system. Note that it may well be the case that the dynamical system can be simulated; the morphological computation can then be replaced by a conventional computation. Nonetheless, such a virtualization requires a complete description of the physics governing the dynamical system. One important aspect of morphological computation is that we do not necessarily need to understand the complete physics of the system we use for computation; all we need is a phenomenological description of its functional dynamics.

There are still a number of important open problems:

- In our definition of a termination, we compared states and asked whether they are equal. This is not critical as long as these states are discrete; for a continuous system our naïve usage of the equal sign is not satisfactory.
- The restriction to deterministic dynamical systems is, in a strict sense, too narrow. Already in the case studies, we will present systems with stochastic properties. As long as one neglects fluctuations, the dynamics of mean values can often be regarded as deterministic. A valid definition of morphological computation should, however, include systems that are governed by statistical physics.

It may also be valuable to clarify another point that sometimes comes up in discussion. In the end, a computation performed on a digital, electronic device relies on the physics of the hardware and can thus be interpreted as a special instance of morphological computation. However, it is also possible to consider morphological and conventional computation as constituting a dichotomy. According to this view, the main difference between morphological and conventional computing is that conventional computers are designed to minimize the influence of the hardware on which they are implemented, ensuring that programs can be written independently of direct physical considerations; in morphological computing this is not required and is not usually possible.

On these grounds, many researchers define morphological computation in such a way that it has to depend significantly on the physical dynamics of the system on which it is implemented. We agree with this position. The definitions presented earlier allow us to specify what we mean by "significantly." In our formalized definition of morphological computation, we defined the concept of a programmable dynamical system in a very broad sense. Additional restrictions allow us to define subcategories of morphological computation. In particular, we can define the class of non-portable morphological computations. The corresponding requirement is that the program for the computation (in the sense defined above) should not contain all the information necessary to perform a desired computational task on a Turing machine. In this category of morphological computation, part of the information determining the computation is given by the physical structure of the device performing the computation. Non-portable morphological and conventional computation constitutes a true dichotomy that matches intuitive ideas about morphological computation. All instances of morphological computation we discuss here are of the non-portable type.

The formalization we have presented matches the Venice definition of morphological computation, providing a formal interpretation for each of its terms. Although we are aware that our formalization may rule out some systems one would like to regard as performing morphological computation, we believe that it can provide clarification for future theoretical studies.

The first requirement in the definition was that reproducible input should be converted into reproducible output. With the concept of a termination, we can cover a broad range of outputs, including oscillatory movements. We observe that the notion of reproducibility is closely related to the ability of the observer to distinguish states. A morphological computation is regarded as reproducible as long as the outputs cannot be distinguished by the observer. Therefore, an asymptotic transient toward a limit cycle may at some point no longer be recognized as being different from the limit cycle itself. Two such transients may differ in an absolute sense for all times; our framework captures the notion that if they are sufficiently close to each other, they are, from the observer's perspective, identical.

The second requirement in the Venice definition is that morphological computing systems should be programmable. We give a precise definition of what is understood by programming that includes parameter tuning.

Finally, the Venice definition requires that the computations performed by a morphological computing system should have some sort a sort of teleological embedding. In our framework, this embedding is understood and specified as a computational interpretation.

2.2.2 Formalizing Morphological Control

For the purpose of this article, we define morphological control as a control process that relies at least partially on morphological computation. Since we have defined the functional dynamics of systems liberally enough to allow all sorts of terminations as possible outputs, we can regard instances of morphological control tasks as special cases of morphological computation in the absence of an observer or computational interpretation. We should nonetheless stress that there is a basic distinction between control and computation, be it conventional or morphological: A computation maps an input onto an output, with the input completely given at the start of the process. Control, by contrast, generates a stream of output signals from a stream of input signals. The input at any given time depends on previous outputs. Thus, the complete stream of input signals is not known at the start of the computation. As a consequence the relation between input and output is much more intricate than in a computation involving a permanent interplay between morphological and conventional control, as shown in Figure 1. We are currently preparing a more elaborate definition of morphological control that takes account of sequences of input signals. When inputs are discretized, the problem is rather easy from a formal point of view. However, coping with nondiscrete input that is not generically discretized as in a conventional computation offers a considerable challenge.

2.3 Programming Morphological Computation and Control

When we perform a computation, morphological or conventional, we perform it on the representation of a system. It follows that the difficulty (we deliberately avoid the term complexity) of a computation

may well depend on the properties of the state space for the system, and not on the intrinsic difficulty of the computation. A binary representation of a computational task (e.g., a simulation on a universal computer) is a case in point. The intrinsic structure of the computer's state space (ideally, a long sequence of bits) is completely independent of the physics of the process to be simulated. Furthermore, the representation is largely independent of the physics of the device performing the binary computation except for the fact that it needs to provide a sufficiently large set of discrete states and the ability to switch between them. Whether a representation is binary is not of importance as such; what is important is the use of a finite alphabet of some fixed size. It is a key property of modern computer programs that they are *device independent*. This has (at least) two important advantages: first, universality (the ideal computer is an abstract Turing machine whose capabilities encompass everything we deem as computation), and second, portability—the ability to run the same program on different machines. To ensure portability, it is necessary to minimize the influence of the physics of specific machines (their specific *embodiments*).

That embodiment may yield considerable efficiency gains is nothing new to digital computer engineers. In fact, embedded systems quite often profit from wiring topologies that reflect properties of the space to be simulated. For example, FPGA-based customizable machines have been used to simulate spatially heterogeneous chemistries [5, 40]. The embodiment—the morphology given by the wiring of the processors—facilitates computation by making it unnecessary to explicitly encode complete spatial information for the molecules in the simulation. At least part of this information is implicitly given by the position of the processor that hosts the data representing the molecule under consideration. Morphological computation extends this idea by avoiding explicit coding of more general physical information. In the case of a system governed by classical mechanics, one way to achieve morphological programmability is to construct a dynamical system with a parameterized attractor landscape. A (possibly digital) controller moves the system from one basin of attraction to another, without coordinating the details of the movement. The pattern itself is realized and stabilized by the attractor. The parameterization of the attractor landscape provides a certain degree of programmability. Importantly, this form of programmability does not require the programmer to encode the physics of the system, as would be necessary for a controller simulating Newton's laws, but only to know (maybe only roughly) the arrangement of the basins of attraction. The downside of such a program is that its portability is limited: It only works for a given physical system.

What are underlying reasons for these gains in efficiency? We claim that there are at least four:

1. Nature is not susceptible to the problems of numerical analysis.
2. In a morphological computation, the physics is already there; there is no need to encode it.
3. Nature is inherently parallel.
4. Instances of morphological computation are proven to be evolvable.

As far as concerns the first point, it is well known that some problems are numerically more difficult than others and that this difficulty may have nothing to do with the intrinsic properties of the problem. One argument in favor of morphological computation is that the translation of the physical reality into strings adds complexity to the problem. As an example, consider collisions among hard objects. If the objects are rotating, simulating their collision dynamics is demanding, even if they are convex [4]. Morphological computation offers a far easier solution, significantly enhancing the efficiency of the control process.

More generally, conventional computing requires the encoding of the whole of the relevant physics. A morphological control process, by contrast, can exploit the fact that the physical system used to perform the computation already incorporates the physics (the second point). Of course, a conventional computer is also a physical system. However, it is designed in such a way that the computation as such is device independent. We can illustrate this point with the example of a recipe. Today, it is not possible to use a simulation to control the cooking of potato soup. Such a simulation would need to include the

complex chemistry and convection dynamics of highly viscous fluids near phase transitions. However, with loosely standardized pots, potatoes, ovens, and spices, a recipe for potato soup takes only a single page. The recipe qualifies as a program, but one that relies on the embodiment of the cooking process. In this way it avoids the need to program the dynamics under consideration. The EU-funded project MATCH-IT (http://fp7-matchit.eu/) is currently applying a similar idea to chemistry, examining how far morphological computing can be used to facilitate chemical process management.

With respect to the third point, we note that in general the computational cost of simulating the interaction among n particles is $O(n^2)$ in CPU time and $O(n)$ in memory. In morphological computation, by contrast, the inherent parallelism of the interactions means that the time necessary for computation is basically independent of the number of particles. Time is only needed for the preparation of the input; this time usually scales linearly with n. Furthermore, general costs (material, energy) for the production of the particles are also linear in n.

As far as concerns the fourth point, our own existence proves that some kinds of morphological control are evolvable (see, e.g., the discussion about the human leg in [27]). However, our claim goes further than this. The evolvability of a system depends critically on the representation of the control problem. Since physical systems can be constructed such that their behavior is a smooth function of their morphology, evolutionary methods provide a good solution for parameter tuning. Note that this is not necessarily the case for control programs represented by strings of characters. Here slight variations may have a large impact. For a detailed discussion of this point, see [30].

The program for a conventional control device consists of a syntactically correct string of characters representing a sequence of semantically meaningful functions. In contrast, current methods for programming morphological control involve tuning parameters and designing systems so that they can generate a large variety of dynamical structures. Even if further formalization is possible, it is unlikely we will ever be able to construct a context-free language for morphological computing. However, it may be possible to identify primitives for specific classes of system and combine them into larger process chains. Developing a methodology to identify and combine functional primitives is a challenge for future work. To date, we know quite well how to program a number of specific systems, but a unifying view is still lacking.

2.4 Statistical Physics for Morphological Control

The dynamics of robots is governed by classical, deterministic mechanics. The system parameters are positions, angles, and their respective velocities. This is the context in which the notion of morphological control was originally developed. However, one of the goals of our work is to enlarge the scope of morphological control to systems that are subject to statistical physics and to exploit the specific properties of their dynamics. Examples of such systems include chemical reactions, functional many-particle systems (such as membranes), and self-assembling systems. These systems are characterized by diffusive transport and phase transitions—two categories of processes absent in classical mechanics. In this setting, system parameters are no longer (only) positions and velocities, but also chemical and thermodynamic properties such as densities, temperature, or entropy.

In what follows, we will focus on self-assembly in chemical systems, showing how we can use the morphological properties of a system to control a construction process, while simultaneously profiting from statistical physics. In a self-assembly process, two components connect only if their shapes match. On its own, however, this is a necessary but not a sufficient condition. They also have to encounter each other when they are in a proper configuration (e.g., when they have matching orientations). In the realm of statistical physics, a system can operate on timescales where such encounters are highly probable. As previously discussed for the case of asymmetric diffusion [39], shape-dependent interactions and appropriate timescales can play a dynamic role in the control of sorting processes.

We mentioned that in order to take a statistical perspective, it is necessary to adopt appropriate timescales. If diffusive transport plays a role, there is a fundamental relation between scales for time and length, which limits the size of systems that can be used for practical applications (though size is no fundamental barrier [43].) A rough estimate of these scales can be obtained by calculating the

average time t_{diam} to transport an object of diameter d over a distance d. Using $\left\langle (\vec{r}(t_1) - \vec{r}(t_0))^2 \right\rangle = 6D(t_1 - t_0)$ for the expected squared distance traveled by a particle immersed in a fluid with diffusion constant D during a time interval of length $t_1 - t_0$ and the Einstein-Smoluchowski relation $D = k_B T/(3\pi\eta d)$, we obtain

$$t_{\text{diam}} \approx \frac{\pi\eta d^3}{2k_B T} \tag{2}$$

(viscosity is denoted by η). For example, in the case of a particle of a diameter $d = 1$ μm, we have $t_{\text{diam}} = 0.4$ s.

One could argue that this limit to length scales and timescales is a somewhat negative statement: In practice, exploiting the combination of statistical physics and morphology is only possible on small length scales. However, there is also a positive perspective. Not only is morphological control based on statistical physics possible only at small length scales, it is probably the only possibility for control. As demonstrated by Rothemund, who introduced DNA origami, our present technology does not allow molecule-by-molecule construction [34].

Today, programming morphological control in the context of self-assembly consists of choosing appropriate shapes and/or selective linkers (e.g., DNA strands). Self-assembly, however, offers more opportunities for the application of morphological control than just matching linkers. One possibility is hierarchical self-assembly, a process in which a first set of building blocks assemble into compounds, which then become the building blocks for a second-order self-assembly process (more levels can be added). This kind of multi-order process may be difficult to set up. However, it also offers advantages. In particular, we interpret the process as implicit, morphology-determined quality control: Only properly assembled building blocks will match on the next level of self-assembly.

A second opportunity offered by morphological control is the possibility of combining self-assembly with evolvable hardware. In previous work, one of the authors (R.M.F.) demonstrated that a combination of genetic algorithms, self-assembly, and a novel scheme for evolving test problems can generate scalable multipliers [6]. This was possible because parts of the problem were solved implicitly by the geometry of self-assembling digital components.

3 Case Studies

In what follows, we present three case studies illustrating applications of morphological control and the relation between these applications and the definitions developed earlier in this article.

3.1 Tensairity-Based Support Systems for Patients with Movement Impairments

3.1.1 Motivation

One effect of aging is the gradual loss of control over complex movement patterns. This loss may have many causes, one of them being reduced sensory and neural performance. The notion of morphological computation suggests another possible explanation. Aging changes the mechanical properties of the body, altering its attractor landscape and thereby reducing the body's ability to contribute to control tasks. Viewed from this perspective, the difficulty many elderly people experience in controlling their movements is not due (at least not exclusively) to poor performance by their brains, but to changes in their body that make the morphological control problem harder. In what follows, we describe the use of a novel technology, *tensairity* [19], to reshape the body's attractor landscape so as to restore morphological control. The system we have in mind is not some kind of machinery that physically moves the patient as though he or she were a passive object. Rather it modulates the patient's own actions.

3.1.2 Implementation

The term tensairity brings together tensegrity with air; inflatable elements play a crucial role. Tensairity has made it possible to build inflatable bridges that can carry a car using pressures of only

300 mbar. More recently, it has been demonstrated that tensairity can also be used to implement actuated structures [18]. In therapeutic contexts, such structures offer a variety of benefits, including low weight and cost, high intrinsic safety, and high adaptability. However, their softness means they cannot be programmed in the same way as, say, an exoskeleton.

Work by one of the authors (K.J.H.) has shown that the tensairity actuator systems can be used to provide soft support to patients undergoing rehabilitation following stroke, spinal cord injuries, and other conditions leading to partial paralysis. The idea is to provide additional support during standing up, standing, walking, and sitting down. Similar techniques can also be useful for elderly people with weak lower-limb musculature.

Several systems to support walking are already in widespread use for clinical rehabilitation. One such system is the Lokomat (Hocoma AG, Volketswil, Switzerland), a device that combines a treadmill with powered, rigid walking orthoses and an overhead body-weight unloading system. The G-EO System (Reha Technology AG, Olten, Switzerland) offers an alternative approach based on endpoint control: The patient's feet are attached to an external powered actuator system that allows simulation of walking on the level, upstairs, and downstairs. Other powered exoskeleton devices to support walking are under development (e.g., eLEGS, Berkeley Bionics, California, USA; ReWalk, Argo Medical Technologies, Haifa, Israel).

Compared to these systems, tensairity actuators offer several potential advantages. The key differentiating feature is that the actuator provides lightweight soft support to patients suffering from partial paralysis or muscle weakness due to injury, disease, or aging. The idea is to supplement joint supports and orthoses with customized tensairity actuators, or to integrate the actuators into wearable garments. The actuators would not replace the action of the patient but would help the patient to regain control over his or her movements.

To evaluate the potential of tensairity actuators, two of the authors (K.J.H., A.D.) have constructed and tested a biomechanical prototype system that mechanically simulates the human thigh, calf (shank), and knee joint, allowing the integration of different tensairity support actuators (Figure 3). The actuators themselves were custom designed and manufactured to fit the mechanical construction. They were then positioned on each side of the artificial knee joint, and the joint angle was measured using a digital encoder. This setup made it possible to implement a feedback control system so that the knee-joint angle could follow arbitrary, prespecified trajectories (Figure 4). The feedback controller drives a three-way valve, which bidirectionally supplies air to the actuators. In a real-life implementation, the feedback system would take its target angle trajectory from values corresponding to physiological knee-joint motion during functional tasks such as standing up or walking.

Tests showed that the system can lift a 3-kg mass at the notional ankle joint (i.e., at the bottom of the shank) from the vertical to the horizontal, using a pressure of less than 1 bar. The torque generated in this way is 15 N m. This suggests that the system could provide useful joint support at low cost, low weight, and low pressure. The next phase of the study will investigate combinations of controlled

Figure 3. Biomechanical test bed simulating the knee joint, thigh, and shank. The knee joint is supported by a pair of feedback-controlled tensairity actuators. The picture shows a knee-joint extension sequence (left to right).

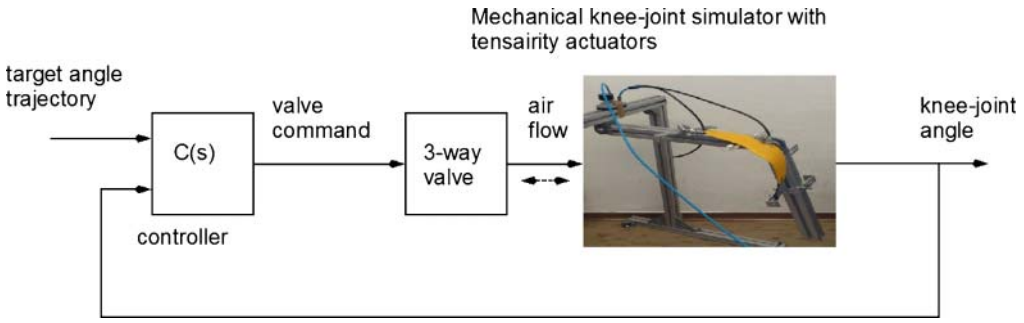

Figure 4. Feedback control system for knee-angle trajectory tracking, with tensairity actuators supporting the knee joint.

tensairity actuators with existing or custom-built joint supports and orthoses. The resulting systems will be tested first with healthy people and then in people with lower-limb impairments.

3.1.3 Interpretation in Terms of the Formal Framework

In terms of the concepts introduced earlier, the patient's body is a compliant, multi-joint mechanical system, which we can consider as a programmable dynamical system $\Pi = (S, M, (f_j)_{j \in J})$. The monoid M represents time and is equal to $\mathbb{R}$. The set S represents the states of the body, given by geometry as well as internal physical parameters such as muscle tension. The terminations one seeks are self-stabilizing gait patterns that are stable on slightly uneven terrain. The transition functions f_j represent the morphology-dependent temporal dynamics of the body; the index j refers to the different morphologies a human being can distinguish and attain. The function of the tensairity actuators is to alter the dynamics of at least one morphology so that the functional dynamics leading to the desired termination becomes as stable and easy to enter (large basin of attraction) as possible.

As mentioned earlier, movement control is the result of the interplay between morphological control and, in the case of a human being, the control exerted by the brain on the body. The calibration of the actuator has to take into account of the patient's physiology as well as his or her neural control capabilities.

3.2 Programmable Self-Assembly of Spatially Heterogeneous Microreactors

3.2.1 Motivation

The synthesis of complex branched molecules (e.g., oligosaccharides for glycol conjugates with medical applications) is still a demanding task. One of the important tasks is controlling potential side reactions [16]. Whereas the synthesis of a linear chain molecule can happen in a template-based manner and the reaction, at least in theory, leads to an unambiguously defined end product, no comparable technique is available for branched structures. Without a template, the only way to achieve unambiguously defined end products would be to use addresses specific to each connection. Unique addressing based on strands of DNA or related chain molecules has previously been used to construct large supramolecular structures [34, 41] and even for the construction of self-assembled compounds of vesicles [8]. However, large linkers are not always suitable for the connection of small molecules. Using small linker structures means that, instead of being able to exploit a combinatorial variety of addresses, one has only a small number of linkers at hand, and that it may be necessary to use the same linker more than once. With one-pot reactions, the multiplicity of reaction pathways lowers the yield [16].

3.2.2 Spatially Heterogeneous Microreactors

The synthesis of branched molecules is a control problem. Given the many types of chemical bond available in the lab (e.g., so-called click chemistries [38]), building arbitrarily branched structures is not a

significant problem in itself. The challenge is controlling the reaction pathways. This is one of the questions addressed by the European research project MATCHIT (Matrix for Chemical IT), coordinated by Steen Rasmussen at The Center for Fundamental Living Technology (FLinT), University of Southern Denmark. The work presented here was funded by the MATCHIT project. For implementations of spatially resolved chemistries, readers should consult the MATCHIT Web site http://fp7-matchit.eu/.

In our study, we demonstrated in silico [32] that it is possible to create a spatially heterogeneous reaction environment with a bias toward specific reaction channels that increases the yield from the reaction (for details of the rather intricate implementation, see [32]). Importantly, this increase is purely due to spatial organization and happens without any kind of interference with the kinetic constants for the reactions. Briefly, we combined two self-assembly processes: First, we used self-assembly to construct a two-dimensional microreactor from what we call *chemtainers* (see details below). Second, we used the microreactor as a platform for the self-assembly of branched oligomers from monomeric building blocks. In what follows, we will limit our discussion to the spatial structure of the microreactor and the way we exploited it to modulate the self-assembly process. Other important aspects of our work (such as the methods used to transport molecules from one chemtainer to the next and the implementation of selective linkers between chemtainers) are not discussed here, but are investigated in the MATCHIT project by other groups.

The self-assembly processes we employed both use selective linkers. Both are reversible (the importance of reversibility is discussed in [42].) Our primary aim was to synthesize branched goal structures as shown in Figure 5. Goal structures are composed of monomers, each equipped with up to three linkers. As described in the figure caption, we used three types of linker. Whether or not the bonds determined by these linkers can be established depends on chemical conditions in the environment, such as the presence of catalysts and whether the linkers match. The reactions required to match linkers take place in chemtainers: potentially microscopic reaction containers that can be linked to other chemtainers to build spatially heterogeneous reaction environments. In the first self-assembly process, the chemtainers self-assemble to form the reaction environment for the second self-assembly process, namely the formation of the branched molecules (the goal structures).

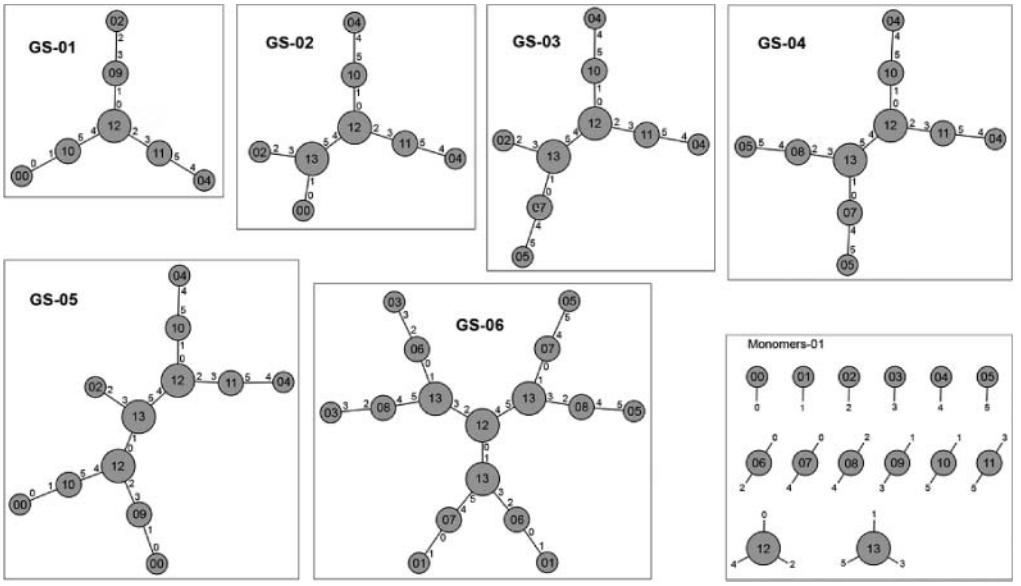

Figure 5. Various goal structures synthesized in spatially heterogeneous microreactors. Each structure is composed of some of the building blocks (monomers) given in the inset in the lower right corner. A monomer is equipped with up to three linkers, making it possible to establish a link between linker pairs 0-1, 2-3, and 4-5. During synthesis, monomers or compounds of monomers can be coupled, as long as a pair of linkers is matching and the chemical environment is such that the coupling can be established.

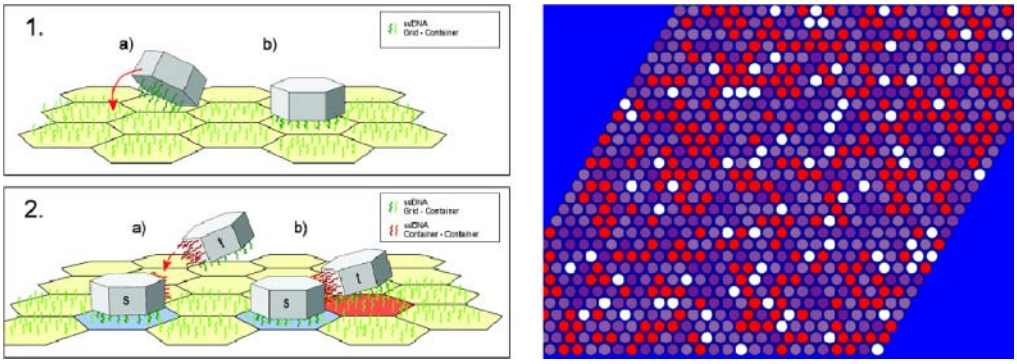

Figure 6. Left: Self-assembly of the 2D grid of containers via specific linker molecules (ssDNA). The upper part (1) shows the association of a container to the grid; (2) illustrates the specific association between two containers (*s*, *t*). Right: Example of a spatially heterogeneous microreactor. In the electronic version, a circle indicates a single container, and the colors indicate the different container types.

To obtain a spatially heterogeneous reaction environment, we assumed hexagonal chemtainers of various types, whose type was specified first by their internal chemistry (to be discussed below) and second by selective linkers on their edges (e.g., DNA-based addresses). The chemtainers self-assemble on a two-dimensional substrate. Their bindings to the substrate are nonselective and relatively weak. Where they are available, bindings to neighbors are reversible and selective. The chemtainers are modeled as freely floating above the substrate, binding to the ground and to already bound adjacent chemtainers. For an illustration see Figure 6. Importantly, bonds between chemtainers are reversible: A bound chemtainer can be released again. If a chemtainer is connected to six matching neighbors, the total binding energy is tuned so that release is no longer probable. The (reversible) self-assembly process is run long enough to ensure the emergence of a structure with a defined neighborhood correlation, though not a spatially regular pattern (Figure 6, right).

Different types of chemtainers provide chemical environments capable of catalyzing the establishing or breaking of some of the possible links between the molecular monomers depicted in Figure 5. Monomers and oligomers are assumed to diffuse between chemtainers. However, the functionality of the chemtainers is maintained. One way of realizing this is by anchoring catalysts to the walls of the chemtainers. If these walls are scaffolds with sufficiently large holes, oligomers can be transported via ordinary diffusion.

If one uses only a small number of different types of linkers (a realistic assumption in glycochemistry), then the chemical functionality of the chemtainers is determined exclusively by the type of link and not by the molecules that are linked. However, it is also possible to link two oligomers if their respective linkers match. This is what makes it so difficult to control the process: The limited number of linkers means that the combination of intermediates (oligomers or monomers) is not sufficiently specific, and therefore many side reactions are possible.

In a one-pot reaction, all oligomers may be connected, always provided that they are equipped with matching linkers. In our system, however, the spatial arrangement of the chemtainers implicitly biases some reaction channels, leading to an increase in yield. For example, our method offers a tenfold increase in yield for the synthesis of oligomer GS-04 (see Figure 5) In Figure 7 we show the results of an evolutionary optimization of the properties of the chemtainer. As explained above, chemtainers are characterized first by their linkers and second by their functionality. It was these properties we attempted to optimize. For each set of parameters, we performed 20 runs, each simulating the self-assembly of the spatially heterogeneous microreactor and the subsequent synthesis of the goal oligomer (GS-04). We then compared the results with those for a one-pot reaction. Figure 7 shows the average yield (as a percentage of the total number of molecules synthesized) and the yield of the best run. Interestingly, the best result was not far from the average. This indicates that our procedure is rather robust.

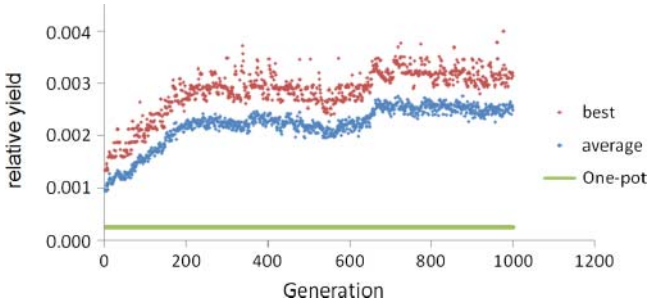

Figure 7. Development of the yield of the goal structure GS-04 during 1,000 generations of the evolutionary algorithm. The lower line indicates the expected yield of polymerization in a homogeneous reaction environment. Chemtainers are characterized first by their linkers to other chemtainers on their edges, and second by their chemical functionality, namely, the ability of their content to influence the links between the building blocks of the oligomers to be synthesized. We evolved these properties using an evolutionary algorithm. For each combination of properties, we performed 20 complete simulations of the self-assembly process.

The arrangement of the chemtainers biases certain reaction pathways by enhancing the probability of some sequences of additions or removals of parts of an intermediate molecule. As an example, consider a chemtainer with functionality A, surrounded by chemtainers with functionality B and C, but not D. When an intermediate molecule M reaches chemtainer A, it undergoes reactions in the order A-B or A-C but not A-D, except for the case in which M travels through chemtainers B or C unaffected and somehow reaches a chemtainer with functionality D. In a spatially homogeneous environment, it would be possible to obtain the same effect by varying kinetic constants and reaction rates with respect to time, perhaps by transferring intermediates from one chemical environment to another. However, this transfer would require external control. The system we used here makes it possible to remain homogeneous with respect to time and to delegate the creation of specific, temporarily ordered sequences of reaction environments to the interplay between diffusion and the patterned reaction environment.

3.2.3 Interpretation in Terms of the Formal Framework

It is well established that self-assembly can be used in order to implement computation [21]. Independent of the interpretation of the outcome, we thus regard self-assembly as an instance of morphological control. This is because it is the dynamics of the system itself that defines the outcome, which is potentially highly specific. The decision of whether two components can be linked is determined locally by the morphology of the linkers without any reference to external control.

In terms of our formalism, the solution containing the chemtainers constitutes a programmable dynamical system $\Pi = (S, M, (f_j)_{j \in J})$. Again, the monoid M represents time and is equal to $\mathbb{R}$. The set S stands for the arrangement of chemtainers on the substrate, the concentration of chemtainers in the solution, and the content of the chemtainers. The transition functions f_j represent the dynamics of the solution, which are basically determined by the choice of matching linkers, the relative concentration of chemtainers in the solution, and the initial content of the chemtainer.

It must be emphasized that in this case, we need to extend the definitions given in Section 2.2. In the stochastic system we consider here, the input is the distribution of chemtainers at the start of the self-assembly process; the program is given by the choice of the chemical functionalities of the building blocks and the structure of their linkers; the terminations are an arrangement of chemtainers with given neighborhood correlations or a given distribution of oligomers (subject to fluctuations). The role of the observer is a trivial one; his or her only role is to run the experiment.

3.2.4 Potential Application

What is the interest of self-assembled microreactors? In the last analysis, a tenfold increase in yield is interesting but not spectacular. One possibility is that our technique could allow us to implement

the production of chemicals on a very small scale. Perhaps more significantly, we hypothesize that spatially heterogeneous microreactors could provide opportunities to use types of catalysts not used in conventional process management. Assume that one does not synthesize oligomers, but some sort of catalyst that is used in a subsequent step. It may well be the case that the environment in which this catalyst is synthesized (or activated) is different from the environment in which the catalyst has to act. In a macroscopic laboratory, this implies that the catalyst has to be transferred from one environment to the other, a process that takes time. This means that catalysts have to fulfill two requirements: First, they have to be efficient, and second, they have to be stable. On a microscopic length scale, where transport involves diffusion over micrometers, 10 s is quite a long time. This means that in microscopic reactors, catalysts, if they are produced in situ, only have to be efficient but no longer need to be particularly stable (an average lifetime of minutes is sufficient). As a result, many more molecules qualify as potential catalysts. The possibility that microreactors could enable the use of metastable catalysts in a technically feasible manner is more than sufficient justification for the study of such systems.

3.3 Multi-Scale Approaches in Systems Medicine and Oncology

3.3.1 The General Systems Perspective in Medicine

Systems approaches are becoming increasingly popular in applied oncology [3] and other medical settings [1]. Today, many authors combine a molecular perspective with approaches that rely on the dynamics of intracellular structures on larger size and timescales or that rely on the way whole cells are organized in tissues. The list of cellular processes relying on the interplay of molecular kinetics with physical aspects of mesoscopic structures such as membranes [33] includes endocytosis, aspects of the cytoskeleton, and macromolecular crowding. The way the physics of the cytoskeleton interacts with cellular information processing is discussed in [12] and [13]. Interpreting the behavior of cells, aggregates of cells, and whole tissues in terms of a dynamical system with chemical and physical components acting on different length scales and timescales opens new perspectives on phenomena that are already familiar in medical practice.

If a biological system is regarded as a network of interacting components, the network constitutes a dynamical system. Such a system can misbehave for one of two fundamentally different reasons: Either (1) one or several components cease to operate in the way necessary to maintain the overall functionality of the network, or (2) the individual components continue to operate correctly, but the network as a whole exhibits undesired dynamics. There are many therapeutic measures for tackling medical problems due to the failure of system components, which we can repair or replace. However, dysfunctions caused when an otherwise healthy system exhibits pathological self-stabilizing dynamics (when it falls into the wrong basin of attraction) are less well understood. One possible example is ventricular fibrillation.

Possible therapies can again be divided into two groups. The first group aims at pushing the system from the wrong into the right basin of attraction (as probably occurs in defibrillation). The second group tries to reduce the probability of accidental transitions from one basin into the other. From the perspective of morphological control, the existence of these basins of attraction is no accident, but the result of an evolutionary process that benefits from this form of control scheme (probably in combination with others that rely purely on chemical kinetics). We argue that concepts from morphological control can contribute to the design of novel therapeutic strategies, especially for dysfunctions of the wrong-basin-of-attraction type. Possible strategies could combine molecular with physical measures.

3.3.2 An Example of the Systems Perspective in Oncology

In radio-oncology, tumors are treated by irradiating the tumor tissue with ionizing radiation. In addition, moderate heating (up to 43°C) of the tumors (hyperthermia) results in a synergistic cell-killing effect. Interestingly, heating alone (below 45°C) does not kill cells. The mechanism behind

the synergistic effect is poorly understood, but it is known that heat and ionizing radiation both have an effect on a wide variety of intracellular processes and structures as well as on the whole tissue. Most existing radiobiological models cannot really explain the response of the tumor system to hyperthermia and radiotherapy. Most prominently, such models do not properly explain the nonmonotonic dose-response behavior of the tumor cells in the case of low-dose hypersensitivity (LDHS). One would assume that if low doses of ionizing radiation kill tumor cells, increasing the dose would produce a larger effect. However, this assumption is not corroborated by the experimental results (see Figure 8).

We start from the observation that tumor dynamics are well described by a class of rather simple models, based on a quantity Γ, called the *dose equivalent* [37]. Importantly, this quantity is an emergent property of the cell. The model template describing the structure of the class of models is given by a set of differential equations:

$$\begin{aligned} \frac{dN_i}{dt} &= f(N_i, N_k, \ldots, \Gamma) \\ \frac{dN_k}{dt} &= g(N_i, N_k, \ldots, \Gamma) \\ \frac{d\Gamma}{dt} &= R - h(\Gamma) \end{aligned} \tag{3}$$

where N_i denotes the number of tumor cells that are not damaged by heat or radiation, N_k denotes the number of tumor cells with lethal or sublethal damage, the dose rate R is the absorbed radiation energy per unit mass, and Γ denotes the dose equivalent. In this setting, the functions f and g determine the rates of damage induction, repair, and cell killing on the population level, whereas the function h describes the (average) repair of cellular damage (changes of the repair capacity of a single cell). Specific choices for the functions f, g, h lead to different types of models, described in detail in [37]. These choices are justified by explicit assumptions about cellular processes, such as the repair kinetics. In previous work, Scheidegger et al. reported at least two types of models (the Γ-LQ and the Γ-IR model) that are able to reproduce biological observations (see Figure 8 for the Γ-IR model as presented in [36].)

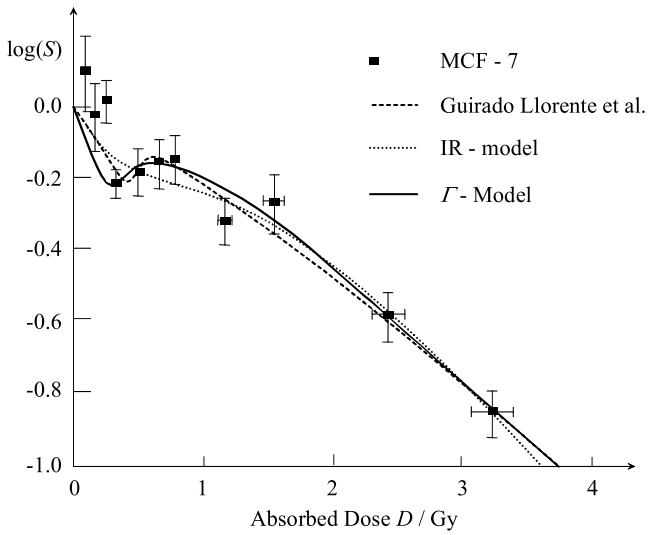

Figure 8. Comparison of the proposed Γ-IR model (Equation 4) with the model of [7], the IR model, and experimental data (irradiated multicellular spheroids of breast cancer MCF-7 cell line). *S* denotes the fraction of surviving cells as a function of the absorbed dose *D*.

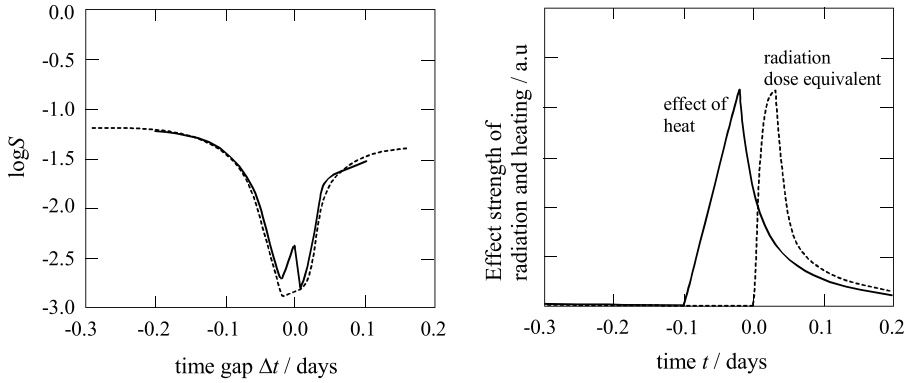

Figure 9. Left: Logarithm of surviving fraction as a function of the time gap between HT and RT fraction (solid: simulation of an HT-RT model according to the model structure of variant 1 above; dashed: results corresponding to the experimental observations). Right: Schematic illustration of the underlying dynamic process; the effect strengths of heat (solid) and radiation (dashed) can be described by a dose equivalent for radiation and the fraction of thermally affected cells for heat.

For the purposes discussed here, it is important to note that the dose equivalent is an input for the functions f, g in the equations. This indicates that the dose equivalent is more than a purely descriptive quantity, and can be used as a true parameter of the system.

Further developments have made it possible to understand the synergistic effect of hyperthermia and radiotherapy (HT-RT model). Following the concept of a dose equivalent, the thermal effect can be summarized by a second, heat-related dose equivalent, which we call Λ. We are currently considering two extensions to our previous system of equations (2) that allow us to model how heat-induced denaturation of proteins can contribute to or interfere with cellular repair:

- Variant 1:

$$\frac{dN_i}{dt} = f(N_i, N_k, \ldots, \Gamma, \Lambda), \qquad \frac{dN_k}{dt} = g(N_i, N_k, \ldots, \Gamma, \Lambda)$$
$$\frac{d\Gamma}{dt} = R - b(\Gamma), \qquad \frac{d\Lambda}{dt} = u(\Lambda) - w(\Lambda) \tag{4}$$

- Variant 2:

$$\frac{dN_i}{dt} = f(N_i, N_k, \ldots, \Gamma), \qquad \frac{dN_k}{dt} = g(N_i, N_k, \ldots, \Gamma)$$
$$\frac{d\Gamma}{dt} = R - b(\Gamma, \Lambda), \qquad \frac{d\Lambda}{dt} = u(\Lambda) - w(\Lambda) \tag{5}$$

Up to now, it is not clear which of the two variants offers a better explanation for the behavior we observe in Figure 9.

The observation that a low-dimensional parameterization is capable of explaining the dynamics of a system at least to some degree of accuracy brings up three questions:

1. First, why is it possible to map a generically high-dimensional system to a low-dimensional one?
2. Second, what is the interpretation of the observed parameters?
3. Third, how can we interpret and control the dynamics of the observed parameters?

With respect to the first question, the engineer's perspective offers a hypothesis for the explanation of the low dimensionality. Biological systems have evolved to be robust, and, at least in technology, robust behavior is much easier to realize in low- than in high-dimensional systems.

Before answering the second question, one should consider a mathematical fact: A single curve (or higher-dimensional attractor) can be parameterized in many ways, using coordinates that may or may not have an obvious physical foundation. In the case discussed here, the dose equivalent may be justified using considerations of statistical mechanics and cellular structures. Since a cell is not an ideal gas or crystal and there is no well-defined equilibrium, the exact calculation of the dose equivalent seems to be impossible, and we treat it in a phenomenological manner. But though our present understanding of the dose equivalent is incomplete, this does not affect its usefulness as a parameter. Given that we have discovered a parameterization that effectively describes cellular dynamics, the second question addresses the task of finding a physical interpretation for the parameters—for example, of explaining them in terms of appropriate system-specific functional primitives.

The third, quite different task is to explore the attractor landscape in terms of the observed parameterization: We hypothesize that the attractor landscape of the tumor system is related to the malignancy of tumors and also to its response to cancer treatment.

The genetic instability of malignant tumors leads to a diverse collection of a number of subclones, each of which is equipped with a slightly different, mutated genome. It is at least plausible to assume that the corresponding biochemical attractor landscape resulting from the mutated genomes is less well suited for controlling cell function and that the barriers between different attractors are lowered. In consequence, it may be more difficult to trigger a specific cellular process—a possible reason for the low efficacy of so-called magic bullets (cancer treatments using a specific key molecule). In addition, the fact that a malignant tumor contains subclones with differing attractor landscapes renders the task of finding a specific magic bullet even more demanding. By contrast, the success of treatments combining ionizing radiation and heat could depend on a system-level effect on the whole attractor landscape, not relying on details of specific attractors.

These dynamics, connecting different length scales and timescales, form the dynamical system we are trying to control. In this setting, the terminations are cellular processes, ranging from the synthesis of linear polymers through protein sorting, to cell division. However, there is no observer in the sense of Section 2.2 and consequently no input. The role of the observer is interpreted as an intervention in an otherwise autonomous system.

However, the cellular dynamics that interest us do not depend just on molecular kinetics, but also on the physics of mesoscopic structures, which combine the mechanics of soft matter with elements of statistical mechanics. This means that our current understanding of cellular processes is not sufficient to establish a connection to each of the terms in our formal definition of morphological computation. More specifically, the determination of program and input would require the identification of *cell process primitives* that could be taken as building blocks for the construction of larger process chains. Today, however, we lack a complete list of such building blocks and their potential combinations.

4 Summary and Discussion

In this article, we have discussed the concept of morphological computation and control, a concept already in use in robotics and related fields. The case studies we present show that the concept of morphological computing can easily be extended from classical mechanics to systems governed by statistical physics and chemistry. They also suggest that exploiting the physical dynamics of a system can lead to major improvements in computational efficiency (reduced use of costly resources), not least because it is no longer necessary to encode all the relevant physics. Of course, there is a price to pay: Morphological programs are not portable, and morphological computing devices are not universal. In some cases, they may also offer less precise control than classical control systems. However, precise control may not be necessary so long as the qualitative aspects of the dynamics are sufficiently robust against perturbations.

These advantages and disadvantages are illustrated by the case studies. In the first study, we are using morphological control to change the dynamics of a physical system (the patient's body) in such a way that it is easier for the patient to control. Experiments planned for the near future will test the effectiveness of this approach. If it is shown to be valid, it should be relatively easy to further improve the stability of the control mechanism, to make it easier to use, and to use it as a complement to existing therapeutic schemes.

Recent results indicate that there is a subtle interplay between the dynamics of the body and sensory functions [9, 31], especially for the elderly. The theoretical framework presented by Hauser et al. [10] goes far beyond the simple picture of a body as a dynamical system with different attractors, describing a feedback system in which an electronic control unit (a feed-forward neural network) and a body constitute a feedback system. It may be possible to use such a system to test learning strategies, and use the results to design more efficient therapeutic strategies.

The second case study discusses morphological control in the context of chemical self-assembly. In the case we describe, it is the morphology that controls which building blocks are connected, a process that is technically too difficult to be executed by a central control unit. This, however, is a relatively trivial example of morphological control. Future studies should also consider dynamical forms of self-assembly, which can form self-healing systems (see, e.g., http://www.math.udel.edu/MECLAB/Projects/SelfAssembly/selfassembly1.htm). In this kind of system, which is also discussed in [26], removing components only temporarily limits the functionality of the system. The dynamic self-assembly process and the morphology of the building blocks allow it to reconstruct itself. This is a good example of a kind of dynamics that is not covered by our formalization of morphological computation and control in Section 2.2 and points to the need for a more extensive definition.

The third case study suggests that the concept of morphological control can help us to understand the response of tumors to treatment or, more generally, to understand the dynamic response of the human body, organs, or tissues. The idea of implicitly encoding a behavior instead of encoding details of cellular processes explicitly leads to the concept of summarizing radiation-induced or thermal damage by dose equivalent quantities. First results in the field of radiation oncology are promising. This suggests it could be useful to focus future research on the development of adequate but accessible descriptions of the dynamics of biological systems.

Taken together, the case studies and the theoretical definitions proposed earlier suggest the existence of a hierarchy of control:

1. *Conventional electronic control*: Based on universal computers and string manipulation; precision limited only by the size of available memory; can be implemented on many different classes of hardware; no implicit robustness; need to handle physical noise.
2. *Systems in which parts of the program of a conventional computer are replaced by a physical system*: Domain restricted (no universal computation); precision restricted (the properties of the physical part may be subject to changes, e.g., elasticity may depend on temperature); only one implementation; implicitly robust; implicitly resistant to noise.
3. *Systems in which conventional computers and physical systems mutually interact*: For a discussion see [11].
4. *Purely non-conventional computation and control*: No classical control involved. Examples are almost all biological information-processing systems.

For engineering, the most interesting systems are those belonging to classes 2 and 3. Investigations aiming at a better understanding of biological systems and their potential dysfunctions will mostly deal with controllers in class 4, taking into account that morphological control can be, and in actual biological systems is, nested. Nesting here means that cell function and the way cells form tissues and organs are both subject to morphological control.

In principle, we can extend this hierarchy further, by taking account of learning processes. In this framework, learning may involve changes in the program of the conventional controller as well as in the morphology of the body. However, a mathematized theory of learning that combines abstracted and physical aspects of control is not yet available. Work by Thomas Schack and his collaborators in the EU-funded project AMARSi (http://www.amarsi-project.eu/) promises a conceptual, theoretical, and experimental framework applicable to human motor skills, but also in a more general setting [35]. It has been claimed that such a theory could have useful applications in physiotherapy and rehabilitation.

To conclude, our case studies suggest that morphological computing and control have the potential to open up new application domains where conventional computing solutions would be computationally expensive or infeasible. To realize this potential, it will be necessary to turn what is basically a set of engineering concepts into a true conceptualization of morphological computation. Obviously, this will require much theoretical and experimental research. In this article, we have attempted to take a step along this road. We make no claim that our formalization of morphological computing and control is complete or that our case studies prove its value. What we hope is that we have demonstrated sufficient progress to justify further research in the future.

Acknowledgment

The work presented in Section 3.2 was supported by the EU FP7 project MATCHIT.

References

1. Ahn, A. C., Tewari, M., Poon, C., & Phillips, R. S. (2006). The limits of reductionism in medicine: Could systems biology offer an alternative? *PLoS Medicine*, *3*(6), *e208*, 710; DOI 10.1371/journal.pmed.0030208.
2. Burtscher, M., Gatterer, H., Flatz, M., Sommersacher, R., Woldrich, T., Ruedl, G., Hotter, B., Lee, A., & Nachbauer, W. (2008). Effects of modern ski equipment on the overall injury rate and the pattern of injury location in alpine skiing. *Clinical Journal of Sport Medicine*, *18*(4), 355–357.
3. Costa, J. (2008). Systems medicine in oncology. *Nature Clinical Practice Oncology*, *5*(3), 117.
4. Donev, A., Torquato, S., & Stillinger, F. H. (2005). Neighbor list collision-driven molecular dynamics simulation for nonspherical hard particles. II. Applications to ellipses and ellipsoids. *Journal of Computational Physics*, *201*, 765–793.
5. Füchslin, R. M., & McCaskill, J. S. (2001). Evolutionary self-organization of cell-free genetic coding. *Proceedings of the National Academy of Science of the U.S.A.*, *98*(16), 9185–9190.
6. Füchslin, R. M., Maeke, T., Tangen, U., & McCaskill, J. S. (2006). Evolving inductive generalization via genetic self-assembly. *Advances in Complex Systems*, *9*, 1–29.
7. Guirado Llorente, D., Aranda, M., Ortiz Seidel, M., Mesa Perez, J. A., Vega Fernandez, J. M. D. L., Martinez Luna, R. J., Zamora Ardoy, L. I., Villalobos Torres, M., & Lallena, A. M. (2010). Low dose hypersensitivity in multicellular tumour spheroids. *Radiotherapy & Oncology*, *96*(Suppl. 1), 607–608.
8. Hadorn, M., & Eggenberger Hotz, P. (2010). DNA-mediated self-assembly of artificial vesicles. *PLoS One*, *5*(3), e9886; DOI 10.1371/journal.pone.0009886.
9. Harry, J. D., Niemi, J. B., Priplata, A. A., & Collins, J. J. (2005). Balancing act. *IEEE Spectrum*, *42*(4), 36.
10. Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R., & Maass, W. (2011). Towards a theoretical foundation for morphological computation with compliant bodies. *Biological Cybernetics*, *105*(5–6), 355–370.
11. Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R., & Maass, W. (2012). The role of feedback in morphological computation with compliant bodies. *Biological Cybernetics*, *106*(10), 595–613; DOI 10.1007/s004-22-012-0516-4.
12. Ingber, D. E. (2003). Tensegrity I. Cell structure and hierarchical systems biology. *Journal of Cell Science*, *116*, 1157–1173.
13. Ingber, D. E. (2003). Tensegrity II. How structural networks influence cellular information processing networks. *Journal of Cell Science*, *116*, 1397–1408.
14. Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 78–80; DOI:10.1126/science.1091277.

15. Kari, L., & Rozenberg, G. (2008). The many facets of natural computing. *Communications of the ACM*, *51*(10), 72–83.

16. Koeller, K. M., & Wong, C. H. (2000). Complex carbohydrate synthesis tools for glycobiologists: Enzyme-based approach and programmable one-pot strategies. *Glycobiology*, *10*(11), 1157–1169.

17. Lichtensteiger, L., & Eggenberger Hotz, P. (1999). Evolving the morphology of a compound eye on a robot. In *Proceedings of the Third European Workshop on Advanced Mobile Robots (Eurobot 99)* (pp. 127–134). Piscataway, NJ: IEEE.

18. Luchsinger, R. H., & Bräker, M. (2010). A novel pneumatic actuator with tensairity. In C. A. Brebbia (Ed.), *Design and nature V*. Southampton: WIT Press.

19. Luchsinger, R. H., Pedretti, A., Pedretti, M., & Steingruber, P. (2004). The new structural concept tensairity: Basic principles. In A. von Zingoni (Ed.), *Progress in structural engineering, mechanics and computation*. London: A. A. Balkema.

20. 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.

21. Mao, C., LaBean, T. H., Reif, J. H., & Seeman, N. C. (2000). Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature*, *407*, 493–496.

22. Marr, D. (1969). A theory of cerebellar cortex. *Journal of Physiology*, *202*, 437–470.

23. McGeer, T. (1993). Dynamics and control of bipedal locomotion. *Journal of Theoretical Biology*, *16*(3), 277–314.

24. Mildner, E., Lembert, S., & Raschner, C. (2010). Influence of ski boots on balance performance. *Sportverletzung-Sportschaden*, *24*(1), 31–35.

25. Paul, C. (2004). Morphology and computation. In S. Schaal, A. Ijspeert, A. Billard, S. Vijayakkumar, J. Hallam, & J.-A. Meyer (Eds.), *Proceedings of the Eighth Conference on the Simulation of Adaptive Behavior: From Animals to Animats 8* (pp. 33–38). Cambridge, MA: MIT Press.

26. Pelesko, J. A. (2007). *Self assembly: The science of things that put themselves together*. Boca Raton, FL: Chapman & Hall/CRC Press.

27. Pfeifer, R., & Bongard, J. (2007). *How the body shapes the way we think*. Cambridge, MA: Bradford Books, MIT Press.

28. Pfeifer, R., Lungarella, M., & Iida, F. (2007). Self-organization, embodiment, and biologically inspired robotics. *Science*, *318*(5853), 1088–1093; DOI:10.1126/science.1145803.

29. Pfeifer, R., Packard, N., Bedau, M., & Iida, F. (Eds.). (2007). *Proceedings of the First International Conference on Morphological Computation*.

30. Poli, R., Langdon, W. B., McPhee, N. F., & Koza, J. R. (2008). *A field guide to genetic programming*. Published via http://lulu.com; freely available at http://www.gp-field-guide.org.uk.

31. Priplata, A. A., Niemi, J. B., Harry, J. D., Lipsitz, L. A., & Collins, J. J. (2003). Vibrating insoles and balance control in elderly people. *The Lancet*, *362*, 1123–1124.

32. Reller, B. (2010). *Programmable self-assembling spatially heterogeneous micro-reactors*. Master thesis, Universität Zürich.

33. Reynwar, B. J., Illya, G., Harmandaris, V. A., Müller, M. M., Kremer, K., & Desorno, M. (2007). Aggregation and vesiculation of membrane proteins by curvature-mediated interactions. *Nature*, *447*, 461–464.

34. Rothemund, P. W. K. (2006). Folding DNA to create nanoscale shapes and patterns. *Nature*, *440*(16), 297–302.

35. Schack, T., & Ritter, H. (2009). The cognitive nature of action—Functional links between cognitive psychology, movement science and robotics. In M. Raab, J. Johnson, & H. Heekeren, *Progress in brain research: Mind and motion—The bidirectional link between thought and action* (pp. 231–252). Burlington, MA: Elsevier.

36. Scheidegger, S., & Füchslin, R. M. (2011). Kinetic model for dose equivalent—An efficient way to predict systems response of irradiated cells. In W. Maurer (Ed.), *Proceedings of ASIM 2011*.

37. Scheidegger, S., Lutters, G., & Bodis, S. (2011). A LQ-based kinetic model formulation for exploring dynamics of treatment response of tumours in patients. *Zeitschrift für Medizinische Physik*, *21*, 164–173.

38. Sharpless, K. B., Kolb, H. C., & Finn, M. G. (2001). Click chemistry: Diverse chemical function from a few good reactions. *Angewandte Chemie, International Edition*, *40*(11), 2004–2021.

39. Shaw, R. S., Packard, N., Schröter, M., & Swinney, H. L. (2007). Geometry-induced asymmetric diffusion. *Proceedings of the National Academy of Sciences of the U.S.A.*, *104*(23), 9580–9584.

40. Tangen, U. (2010). Enzyme-like replication de novo in a microcontroller environment. *Artificial Life*, *16*(4), 311–328.

41. von Kiedrowski, G., Eckhardt, L. H., Naumann, K., Pankau, W. M., Raimold, M., & Rein, M. (2003). Toward replicatable, multifunctional, nanoscaffolded machines. A chemical manifesto. *Pure and Applied Chemistry*, *75*(5), 609–619.

42. Whitesides, G. M., & Boncheva, M. (2002). Beyond molecules: Self-assembly of mesoscopic and macroscopic components. *Proceedings of the National Academy of Sciences of the U.S.A.*, *99*(8), 4769–4774.

43. Whitesides, G. M., & Grzybowski, B. (2002). Self-assembly at all scales. *Science*, *295*, 2418–2421; DOI:10.1126/science.1070821.