Morphology and Computation

Chandana Paul
Artificial Intelligence Laboratory
University of Zurich
CH-8051 Zurich, Switzerland
chandana@ifi.unizh.ch

Abstract

The body of a situated agent plays an important role in adaptive behavior as it enables sensory motor interactions with the environment which can give rise to emergent intelligent behavior. Using the physical dynamics of the body, the agent can perform behaviors with much simpler control than would otherwise be required for the same task. The physical structure of the body, or the *morphology*, determines its possible set of sensory motor interactions as well as its dynamics, and as a consequence, determines the complexity of the control required. This intrinsic relationship between the morphology and required control has been identified as the *morphology and control tradeoff* (7). What has not been previously considered, however, is that the controller is a computational entity whereas the morphology a physical entity, and if a tradeoff exists between them, it may be possible for the morphology to subsume a computational role. This paper introduces the idea that a robot body can be used for computation, in addition to merely acting as an effector for the controller. It shows how this computation may be used to reduce the computational demands on the controller.

Keywords: embodiment, robots, morphology

1. Introduction

It is known in the study of adaptive behavior that the morphology of an agent's body can be used to achieve simplified control. Brooks showed that embodiment would enable the use of simplified controllers requiring little or no representation (2). Pfeifer has similarly discussed the importance of considering the design of the morphology of a robot, to reduce or simplify the control required (7). The Braitenberg vehicles (1) provide an excellent example of this concept. In these vehicles the simple positioning of sensors and motors with hardwired connections between them, can be used to achieve various "intelligent" behaviors. In a more biologically inspired system, Cruse (4) gives an example of how mechanics of the body are used in biology, to simplify the coordination of insect legs. However, although these studies have discussed the reduced computational complexity of control as a result of the dynamics of the body, they have not considered that this may be the case because in effect the morphology performs some of the "computation" that the controller would otherwise perform. The goal of this paper is to investigate this unexplored possibility.

2. The XOR Robot: a thought experiment

Consider a robot which is controlled by perceptron networks. The perceptron is a well-known neural network, which consists of a single output neuron, with multiple inputs, adjustable synaptic weights and a threshold function. The perceptron convergence theorem, proved by Rosenblatt showed that the perceptron could be used to compute any function, as long as it was linearly separable (5). This means that the perceptron could be used to compute functions such as AND and OR, for example, but not functions such as XOR or XNOR.

Consider now that there are two perceptron networks, with inputs A and B. The first one performs an OR function, and the second an AND function. The networks are connected to actuated parts of a simple robot body (Fig. 1). The first network is connected to M_1, which is a wheel at the center of the base of the robot which causes forward motion. The second network is connected to M_2, which lifts the wheel off the ground. Now when A and B are both off, both networks output 0, which means that the wheel is on the ground but does not move, so the robot is stationary. When only A is active, the AND network outputs a 0, so the wheel remains on the ground. The OR network outputs a 1, and makes the wheel turn, thus causing forward motion. When B is 1, the situation is the same, the AND network outputs a 0 and the OR outputs a 1, causing forward motion. However, when A and B are both on, then the OR network causes M_1 to move, but the AND network causes the wheel to lift off so that it no longer touches the ground so the vehicle is once again stationary. The following table summarizes the behavior of the robot in these four conditions.

A	B	Behavior
F	F	stationary
F	T	moving
T	F	moving
T	T	stationary

But this table looks like the truth table of the XOR function! How is this possible?

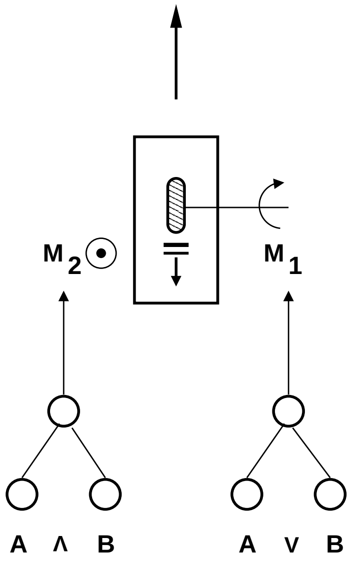

Figure 1: **The XOR Robot** This robot has one wheel, with two actuated degrees of freedom. The motor M_1 is responsible for turning the wheel so that the robot moves forward. The motor M_2 is responsible for lifting the wheel off the ground. Each motor is controlled by a separate perceptron network, which takes as inputs A and B. M_1 is controlled by a network which computes A OR B, and M_2 by a network which computes A AND B. Using only these controllers, the robot is able to display the XOR function in its behavior.

The explanation is that the robot's behavior is not simply determined by the output of the neural networks, but also by the actuated components of the body. The body through its structure provides some additional computational ability, which allows the XOR function to be displayed. In this case, the body provides the computational equivalent of a second layer of neural processing, in which it performs a NOT on its first input, followed by an AND, as shown in Figure 2. The function can be written as $M_1 \wedge \sim M_2 \rightarrow B$. Thus, the example shows that through its configuration a robot body can perform a quantifiable computation on its inputs.

3. The OR Robot

Although at first this seems like a surprising result, it can be shown that this is by no means an isolated occurrance. In fact, various simple morphologies can give rise to computation. In order to show this a second robot example is provided. The OR Robot (Fig. 3) is another simple mechanical structure, with a rectangular body and and two wheels which are aligned along central longitudinal axis of the body, one behind the other. Each wheel has a motor, which when turned on, rotates the wheel so that the body is propelled in the forward direction. When the motor is off, the wheel can also be passively driven.

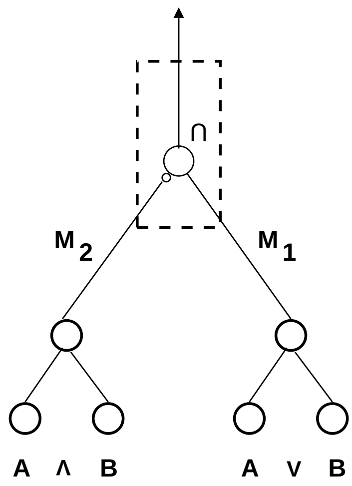

Figure 2: **Computational structure equivalent to the XOR Robot** The body of the XOR robot acts as if it is performing the computational function M_1 AND $\tilde{M}_2$

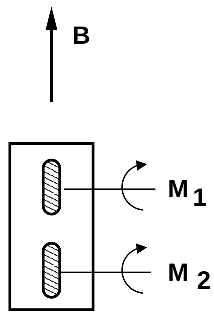

Figure 3: **The OR Robot** This robot has two wheels both of which turn in the same direction when actuated. Each wheel is actuated by one motor which is responsible for turning the wheel. The wheels are also capable of being driven passively.

In this robot, when both motors are off, the robot is stationary. When only M_1 is active, one wheel turns and passively drives the other wheel so the robot moves forward. Similarly, when only M_2 is active the robot, the robot also moves forward. Finally, if M_1 and M_2 are both active, both wheels turn, and so the robot also moves forward. The behavior of the robot in these four conditions is summarized in the table below.

M_1	M_2	Behavior
F	F	stationary
F	T	moving
T	F	moving
T	T	moving

It is clear that this table is the truth table of the the familiar OR operator from Boolean logic. The computation performed by the morphology can be represented using the conventional $\vee$ sign, and written as

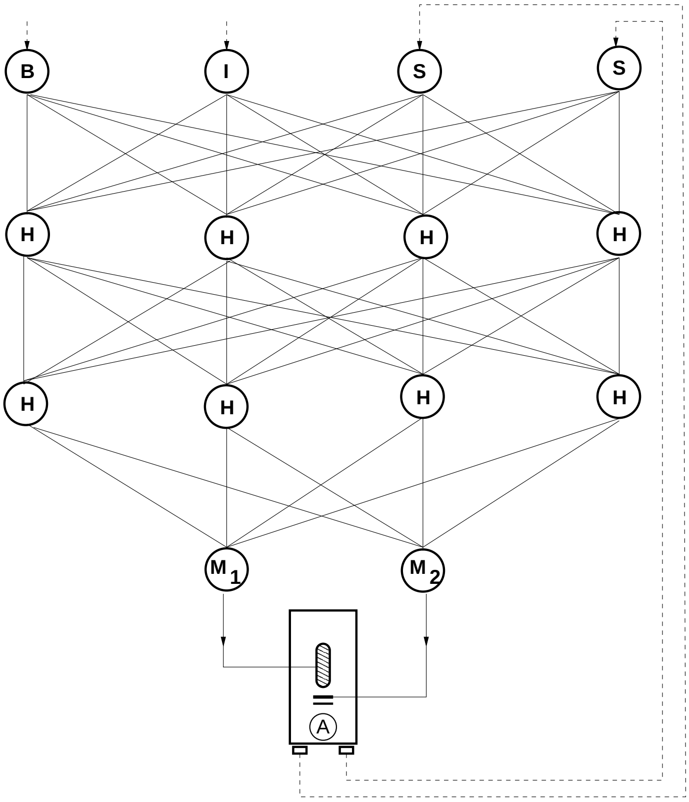

Figure 4: **Conventional neural network architecture for Vaccuum Cleaning Robot** The robot is controlled by a neural network which has an input layer with sensory input nodes, labelled S, input node I and bias node B. There are two hidden layers, each with 4 hidden neurons each, labelled H. The final layer is the output layer which has two output nodes to send motor commands to the robot.

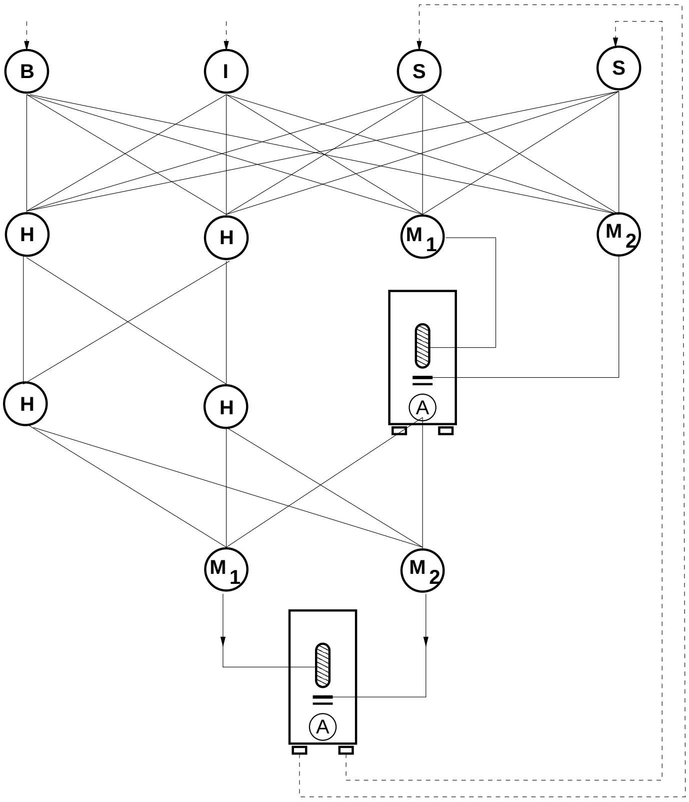

Figure 5: **Morphological computation used in the neural control of the Vaccuum Cleaning Robot** The robot is controlled by a neural network which has an input layer with sensory input nodes, labelled S, input node I and bias node B. There are two hidden layers. The first hidden layer has 2 hidden nodes, represented by H, and two nodes which convey motor commands to the robot, represented by M_1 and M_2. The second hidden layer also has two hidden nodes, labelled H, and a node which conveys the output of the robot accelerometer A. The final layer is the output layer which has two output nodes which send motor commands to the robot.

$M_1 \vee M_2 \rightarrow B.$

4. Vaccuum Cleaning Robot

At first glance, it may be argued that the computation performed by the morphology of the OR and XOR robots described above simply exists in the eyes of the observer, and is not really a computation. However, adopting the common-sense definition that a computation is *real* if it can be used as a computation, it can be shown that the computation is in fact real.

Consider this example. The body of the XOR Robot is now to be used as the basis for a Vaccuum Cleaning Robot (albeit a simple one, which is only suitable for long and straight corridors). It is enhanced with cameras to sense the environment, a microprocessor for computational processing and behavior arbitration, an accelerometer and of course a vaccuum unit. Its task is to survey its initial environment and decide whether to vaccuum or not. If the floor does not look particularly dirty, it can wait. If the floor is dirty, it can decide to start vaccuuming down the long straight corridor. Such behavior arbitration requires a certain amount of processing. A neural network based controller could be used for such a task (to take a particular example of a controller architecture), part of which may look like Fig. 4. This is a standard neural network architecture with an input layer, two hidden layers and an output layer. The input layer receives inputs from the robots visual sensors S, a higher level signal I, and a bias node B. The inputs are processed through three layers of the network and result in output motor commands issued to the robot. It is a familiar computational structure.

However, it could be possible to replace part of the computational structure of the network by the body itself (Fig. 5). The motor inputs M_1 and M_2 could recieve inputs from the nodes of the previous layer, via weighted synaptic connections and the output of the body, as measured by the accelerometer A, could serve as an input for the next layer. It can be seen that if this were done, the network would continue to perform as a computational entity, which simply incorporates the function $M_1 \wedge \sim M_2 \rightarrow B$ into its function. The required movement of the body during the computation would be small, a computational twitch so to speak, as it would last for only two time steps of processing until the final output motor commands were sent. Thereafter the robot could again proceed to use its body for its more conventional purpose of moving down the corridor for vaccuum cleaning.

This example is of course a toy example and quite impractical. Performing the XOR function on a processor would be trivial from the point of view of processing time and energy costs compared to having to move the body. But that is not the point. The point is to demonstrate that the computation can be used as part of a standard computational process, proving that it is in fact *real*.

5. Sensorimotor Control

In the above example, the morphology of the XOR robot served as the basis for the Vaccuum Cleaning Robot. The computation performed by the body of the XOR robot was easy to recognize as a common binary function, and so it could be shown that starting from a morphology which performed a simple computation, a controller architecture could be designed to utilize it. In more complex robot morphologies such as manipulators or legged robots, however, it is not trivial to define the computational function of the body. In robots which have rotary joints actuated by motors and joint angle sensors for sensing, the computation performed is a complex analog function determined by the dynamics of the body. Yet, it can be understood as a computational function nonetheless, in the sense of defining a mapping from motor commands as inputs to the physical consequences measured by the sensors as outputs.

Consider for example a 2 DOF robot manipulator with rotary joints and joint angle sensors, which is controlled by a neural network with sensory feedback (Fig. 6). For simplicity we assume that input I is 0 and the weights of the network are constant. The controller is designed such that it reads in sensory input from the joint angle sensors, processes the information through the network, sends motor commands, and only then reads in the next value of sensory input for the next iteration of neural processing. It is clear to see that in each such iteration there are two stages of computation. One is the transformation of the sensory inputs to motor outputs through the neural network. The second is the transformation of motor outputs to sensory inputs through the forward dynamics of the manipulator. Thus, the entire system forms a computational entity.

Viewing the system from this perspective, it is clear how a morphology and control trade-off can exist in an embodied agent with sensorimotor control. Changing the physical characteristics of the robot, including even simple parameters such as link length or mass distribution, will affect the dynamics of the manipulator. This will, in effect, change the computational relationship between the motor commands and

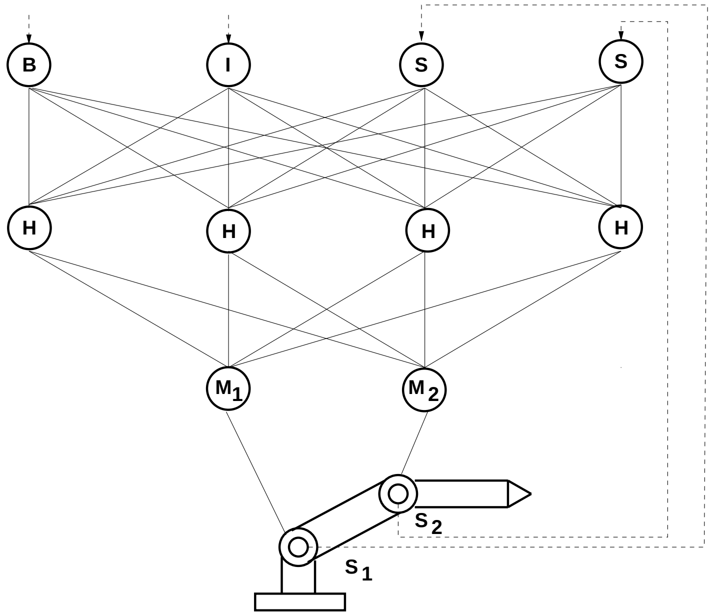

Figure 6: **Sensorimotor control or a robot manipulator** A 2 DOF manipulator is is controlled by a neural network with sensory feedback. The network has an input layer with 2 sensory input nodes, labelled S, an input node I, and a bias node B. The sensory nodes recieve proprioceptive inputs from the manipulator on joint angles. There is one hidden layer with four hidden nodes represented by H. The output layer has two output nodes which send motor commands to the two actuators of the robot, M_1 and M_2

its physical consequences measured by the sensors. Ins some cases this change may be for the worse, but in others it may be for the better with regards to the task, and reduce the requirements on control.

6. Discussion

The previous sections have illustrated the computational potential of morphology, and shown how understanding this potential can elucidate the complex interactions between morphology and control observed in embodied agents. There are many implications of this for understanding natural intelligence, as well as the design of artificial agents. These will be discussed below.

6.1 *Morphological Computation*

The thought experiments of Sections 2 and 3 demonstrate that it is theoretically possible for the morphology of an agent to perform a computational function. Such computation which can be latently performed by the morphology can be called *morphological computation*. The knowledge that physical structure can perform such computation is as old as Babbage's design for his Analytical Engine, a massive brass, steam-powered mechanical computer, from 1837. But as computation has grown more siliconized in the 20th century, a divide has arisen between the computational and physical hardware in robotics and AI. The body is viewed as an effector for the computations performed by the control hardware. While this approach is convenient as it allows for a clear division between mechanical and controller design, it has led to a separation which is somewhat artificial. Moreover, it has occluded the understanding of the interaction between morphology and control in both biology and engineering. The description of the XOR robot serves to remove this occlusion, and show that computational properties can arise even in very simple physical structures. It suggests that it is not simply a rare phenomena, but perhaps a pervasive characteristic of physical structures. Thus, it should be acknowledged and scientifically investigated in the context of robotics if a real understanding of the physical world is to be acheived.

6.2 Explicit morphological computation

The Vaccuum Cleaning Robot demonstrates that when a robot with latent morphological computation is augmented with a sensor which can sense the behavioral consequences, it makes the computational function defined by the morphology explicit, such that it can be used as a standard computational sub-unit, at any stage of processing.

The fact that this is possible shows that morphological computation is not simply a way of *viewing* the transformation of motor commands to physical behavior as a computational process, but that it is a real computation as is traditionally understood, and can be used accordingly.

6.3 Using computational and motor functions

The Vaccuum Cleaning Robot also shows that in a system where appropriate sensing enables the explicit use of morpohological computation, a controller can exploit both the computational and motor functions. This can be acheived by separating the two functions in time, and alternating between them.

6.4 Duality

In engineering, it is traditional to think of a single component of a system performing a single function. Thus, in most cases in human engineered technology each component usually has a single function, and if it has more than one function, it is usually used to perform only one of them at a time. This modular approach to design is the basis of human engineering, which leads to a tacit belief in the idea that natural systems may also be designed according to these principles. However, in the case of an intelligent embodied agent, this does not need to be the case. Although, it is difficult for human engineers to conceptualize, it is possible for a single action of an agent to serve both motor and computational functions *at the same time*. In fact, many physical action performed by an embodied agent have the inherent possibility for *duality of function*.

In order to illustrate the concept, and show that it is relevant for natural intelligence, a simple example from human behavior is given. Consider the case when there are a large number of coins on a desk which must be counted (more than can be counted through a cursory visual inspection) and they must be put away in a drawer under the desk. Notice that there are two components to the task requirements. One is the computational task of counting the coins, which involves the development of an abstract representation (number) based on the environmental condition. The second is a physical task of relocating the coins from the desktop to the drawer. There are multiple ways of accomplishing this, but one common approach to the task that humans use, is to slide the coins off the desktop into the drawer one by one, counting each one in the process. In this simple behavior, the action of sliding the coin into the drawer has duality of function. One function is computational: it serves to separate the coin which has already been counted from the set of coins which has not been counted. The second function is physical: relocating the coins from the desktop to the drawer. These two functions are seamlessly integrated into one action, without any conscious forethought. The example demonstrates that if the opportunity arises to utilize duality of function, natural intelligence has been designed to be able to exploit it.

The Vaccuum Cleaning Robot, however, was an example of "human design" in that the computational role of the body was separated from the physical role in time. This had the advantage that it highlighted the use of the computational function of the body, by separating it from its physical role. However, theoretically, no such separation is necessary. It should be possible to design systems which use their physical structure for both physical and computational purposes at the same time.

6.5 Effect of the environment

The environment plays an important role in morphological computation. In the thought experiments of Sections 2 and 3, the systems were embedded in a single environment. However, when the environment of the system changes, the functions describing the system will also change. This is very clear to see in the examples presented. Consider the case when the OR robot is placed on a downhill incline. Now, the robot moves forward regardless of whether the actuators are active or not. The computational function of the structure changed. Thus, the ecological niche is crucial for the particular characteristics of the morphological computation which arise.

6.6 Morphology and Control

It has been understood in the study of adaptive behavior that embodiment can lead to simplified control. However, the reasons for this have not been clear. It has been known that interaction of loosely coupled sensorimotor processes with the dynamics of the body gives rise to *emergent behaviors* which are not explicitly represented in the controller (3)(6)(7). However, the mechanism of emergence itself has not been further investigated. By showing here that the body can perform a computational role, the basis of such emergence can begin to be understood. In some cases at least it is can be understood that the body provides the “computational glue” between loosely coupled sensorimotor processes.

There is still, however, a long way to go in understanding the relationship between morphology and control. Particularly it needs to be understood what characteristics of the morphology affect control requirements, and how. It is likely that further investigation into this area will require the development of new analytical techniques to identify and quantify the computational contributions of a morphology.

If such an understanding can be acheived, however, it will open a whole new realm of possibilities to design robots which incorporate “intelligence” in their bodies. By utilizing smart morphologies which are known to optimize dynamics for control, more robust and adaptive agents will be able to be designed.

7. Conclusion

This paper introduced the concept of *morphological computation*, computation performed by the mechanical structure of a robot body. To show the existence of such computation, an example of a robot controlled by perceptron networks was given, which utilizing the structure of its morphology was able to display the XOR function in its behavior. The XOR function being linearly inseparable would have been impossible to acheive with only a perceptron network, and thus proves that the morphology can perform a computational role.

This was followed by the description of a vaccuum cleaning robot which used the computation performed by its body in its own neural control. The example demonstrated that the computation performed by the body is both explicit and real. Furthermore, it explicitly illustrated a situation where the morphology replaced part of the computational control. The illustration sheds new light on how smart morphologies could reduce control.

Future work will focus on developing a theoretical framework for morphological computation which will provide analytical methods for studying the computational role of a robot body. This could be used to guide the design of smart mechanical structures, which will be able to exploit the morphology to reduce control.

References

Braitenberg, V. (1984) Vehicles: Experiments in Synthetic Psychology. *MIT Press* Cambrige, MA

Brooks, R.A (1991) Intelligence without Representation. Artificial Intelligence 47, 139-160

Brooks, R.A (1989) A Robot that Walks; Emergent Behaviors from a Carefully Evolved Network, *Neural Computation*, 1(2):253-262

Cruse, H., Bartling, C., Dean, J., Kindermann, T., Schmitz, J., Scumm, M. and Wagner H. (1996) Coordination in a six-legged walking system: Simple solutions to complex problems by exploitation of physical properties. In *Proc. 4th Int. Conf. on Simulation of Adaptive Behavior*, Cape Cod, USA, 84-93

Haykin, S. (1994) Neural Networks: A comprehensive foundation. *Macmillan College Publishing Company*, New York

Mataric, M. J. (1992) Designing emergent behaviors: From local interactions to collective intelligence”, in *From Animals to Animats: International Conference on Simulation of Adaptive Behavior* eds. J-A. Meyer, H. Roitblat, and S. Wilson, MIT Press, 432-441

R. Pfeifer and C. Scheier. *Understanding Intelligence.* MIT Press, Cambridge, Massachusetts, 1999.