IOP Conf. Series: Materials Science and Engineering 1261 (2022) 012011 doi:10.1088/1757-899X/1261/1/012011

Morphological Computation and Control Complexity

Thomas George Thuruthel, Fumiya Iida

The Bio-Inspired Robotics Lab, Department of Engineering, University of Cambridge, UK.

E-mail: tg444@cam.ac.uk

Abstract.

Morphological computation proposes the idea that in a physical system, certain computational processes can be off-loaded to the body. However, the concept has still eluded serious theoretical quantification attempts, unlike traditional computational theory. This perspective examines the notion of morphological computation from the well established theories of traditional computation and computational complexity, drawing parallels between the two, to understand the differences and similarities. Further, we look at the quantification efforts of morphological computation and attempt to link it to the unexplored field of control complexity. We argue that the development of complexity theory for control problems is necessary to study and utilize the concept of morphological computation, if it is possible.

1. Introduction

The notion of *morphological computation* is becoming ever prevalent in the field of robotics [1], especially since the rise of soft robotic technologies [2]. This perspective article examines the notion from the well established theories of traditional computation and computational complexity, drawing parallels between the two, to provide answers to some utilitarian questions; *How does morphological computation theory differ from traditional computational theory? Can morphological computation be quantified? Can physical control problems be classified based on their complexity?*. The organization of the paper is as follows. The first section introduces morphological computation from a pancomputational viewpoint, discusses the properties of traditional computation hardware and field of computational complexity. We conclude with a comparison of morphological computation through the lens of traditional computation. The second section looks into works that attempt to quantify morphological computation for control tasks and their merits. We further propose viable routes to develop the field of control complexity and conclude with a brief discussion.

1.1. What is Computation?

The most general way to look at a computational process is to describe it from a dynamical systems' perspective [3]. To quote [3] directly, "A dynamical system is described in terms of its abstract state space, the system's current state within its state space, and a rule that determines its motion through its state space. In a classical computational system, that rule is given explicitly by the computer program; in a physical system, that rule is the underlying physical law governing the behavior of the system". The pancomputational paradigm (everything is a computing system) is hence adopted by the authors. This implies that any computable function can be represented by a carefully designed dynamical system with an appropriate input-output encoding. Morphological computation, hence, refers to the computation that the body (including the structure, actuators, and sensors) performs while executing a physical task. Note that this is very similar to mechanical computation [4], only differing in their application domain; mechanical computers are designed for solving abstract computational problems whereas morphological computation is associated with solving physical control problems.

1.2. Computational Hardware's and their Properties

The specific dynamical system (hardware) we use for computing depends on the application, requirements, and constraints. The digital electronic computer is the most common amongst all the available artificial hardware systems. They are characterized by transmission of information by discrete and binary electrical signals. This makes them less susceptible to noise, good data transmission bandwidth at low-energy consumption's, efficient data storage and retrieval options, a high cascadability (output of one logic gate can be easily connected to another input) and fan-out (multiple input gates can be connected from a single output gate) due to ease of logic-level restoration and input-output isolation. Note that most of the input-output devices (monitor, for example) are still analog and hence digital computers have to convert information from the digital-to-analog form or vice versa, which is easily doable with electronic circuits.

A close counterpart of the digital electronic computer would be the analog electronic computer, where information is transmitted in continuous electric signals. This would provide them higher bandwidth and faster calculation speeds, but they are easily affected by environmental noise, parasitic effects and a lack of a reliable variable information storage mechanism. Mechanical computers and mechanical analog computers are characterized by information transmission through physical mechanisms (For a detailed review of the field, please refer to [5]). Hence, their computing bandwidth is very low (information travels at the speed of sound when compared to the much faster drift velocity of electrons), highly susceptible to noise and energy inefficient. As expected from a pancomputational viewpoint, there are numerous other computing mechanism like optical computing, quantum computing, DNA computing, etc. Although digital electronic computers seem the best choice for general computing applications, there are scenarios where each type of computing hardware can be better. For example, mechanical computers have been useful in extreme environments [6]; faster and energy efficient computing is potentially achievable in analog electronic computers and optical computers [7, 8, 9]; massive parallelization and large storage is possible in DNA computers [10, 11], and so on. The natural equivalent of a computing hardware, the nervous system, is actually characterized by a poor information bandwidth (the speed of information transfer through a combination of electrical and chemical signals happens at a speed of 70-120 m/s, comparable to mechanical computers [12]). However, these systems have good energy efficiency, massive asynchronous parallelization abilities, compactness and incredible robustness to damage (due to their redundant architecture).

1.3. Task/Problem Complexity

In computational theory, the complexity of a problem is usually measured by the amount of resources required to solve the problem by the application of a particular algorithm. Typically, the resource of concern is time and memory. This formalization allows us to classify problems and provides an estimate of the time and memory requirements imposed on the computational hardware. The Big 'O' notation is the most common metric for calculating how the time complexity of a problem scales with the size of the input. Note that the time complexity of each problem is always relative to the input encoding, however the actual resource utilization will remain the same as the encoding transformation causes a change in input size corresponding to the change in algorithm complexity. In other words, a computational problem cannot be

IOP Conf. Series: Materials Science and Engineering 1261 (2022) 012011 doi:10.1088/1757-899X/1261/1/012011

simplified by just changing the input encoding or the computational hardware. Later we will see that this is not the case for control tasks.

1.4. Characteristics of Morphological computation
Having looked at the digital electronic computation, we can now compare the computational characteristics done through a mechanical system. We assume that physical information travels through the mechanical system purely by mechanical waves, without any signal transduction in between. Although, the notion of morphological computation was introduced to perform non-traditional computation, it is still insightful to predict how it would fare in traditional computational tasks. Natural mechanical systems have smooth and continuous dynamics, thereby making it difficult to isolate their logic networks (or its equivalent). Mechanical systems are also plagued with energy dissipative elements like friction and damping. Both these effects reduces the cascadability and fan-out capacity of mechanical computers (This is not necessarily a disadvantage when it comes to physical tasks, as we will investigate later). The information storage capacity and bandwidth of a mechanical system is more difficult to be analyzed. Even though mechanical waves travel at the speed of sound, which is comparable to the speed of information transfer in the brain (still much slower than the drift velocity of electrons in digital computers), the large loss of data due to damping, friction and poor input-output isolation, would reduce the information processing bandwidth of these systems. Hence, we see that all mechanical systems for computing (traditional) purposes are designed to be discrete by using bistable structures or origami architectures [13, 14, 15, 16]. There have been examples of using continuous mechanical systems for traditional computing, however, the fidelity of the system is unproven and they are still 'programmed' using digital computing hardware's [17].

2. Quantifying Physical Complexity
The field of computational complexity solely focuses on abstract numerical problems. Blum complexity axioms specify desirable properties of complexity measures on the set of computable functions [18]. There are no equivalent axioms for control tasks. Although, there are numerical problems to be solved while performing a physical task, the notion of complexity associated with physical tasks is non-existent to the best of our knowledge. If the complexity of physical tasks can be quantified and classified, the same metrics can be used to evaluate the complexity of the physical body and controller(brain). There have been attempts to quantify morphological complexity and controller complexity that we will analyze and attempt to connect.

A change in dynamical properties of a robot(agent) would affect its control architecture to perform the *same task*. This idea has sometimes been without theoretical evidence been used to put forth the notion of a morphology-controller trade-off [19]. Typically, we tend to associate the complexity of a physical system to its observable degrees of freedom (a piece of metal and silicone can otherwise exhibit the same *levels* of passive deformation) and nonlinear dynamical properties, especially since the rapid growth of interest in soft robotics. The controller complexity is typically linked to the number of active elements, the bandwidth of the sensors and the computational complexity of the mathematical processing involved. A commonly used example that is used to illustrate the 'morphology-controller trade-off' is the universal robotic gripper [20]. Here, the specific design of the robot reduces the number of actuators/sensors required, and the mathematical processing involved in control when compared to the traditional rigid designs for grasping an object. Although, on the surface, this example seems to indicate a reduced *complexity* of the controller, we have to be careful to conclude the same. First, we do not have a good measure to quantify the complexity of the controller. Second, we also need the complexity of the task to be quantifiable and comparable. Although, the universal gripper is good at generating stable grasp of an object, the final objective is still to vary the state of the object in a desirable manner (for example, placing it in a known location in a certain pose) and

IOP Conf. Series: Materials Science and Engineering 1261 (2022) 012011 doi:10.1088/1757-899X/1261/1/012011

this would require more sensing and control elements even with the universal gripper. Another typically omitted detail is that the universal gripper has to be mounted on a robotic manipulator to be able to perform the task.

There have been few works on developing metrics for quantifying morphological computation, and none on control complexity (to the best of our knowledge). A lot of them have focused on the use of physical body to solve traditional computational problems. Our objective is to look into metrics for solving physical tasks. Hence, these works have been omitted. The most promising work on quantifying morphological computation was presented by Keyan Zahedi et. al. [21]. They introduce two concepts for measuring morphological computation using information theory. They state that morphological computation is the effect of the current world state (X) on the next world state (X'), which is not assignable to the controller (C). The world state also includes the internal states of the agent in their formulation. In other words, the world state X contains the internal state of the agent and the states of the task objective (if they are independent of the agent). Based on this, two metrics can be calculated; one that measures the effect of the action on the next world state; the second measures the effect of the current world state on the next world state. At maximal morphological computation, we would expect the first measure to be at its minima and the second to be at its maxima. Information theory measures are then used to quantify these metrics.

The same idea was investigated from a control system perspective in [22]. Here, it was proved that the optimal open-loop controller obeys the relation $I(X';C) = 0$ (not a sufficient condition). This means that the mutual information between the random variables X' and C is zero or that they are independent. An optimal open loop controller would lead to the maximum entropy reduction from the initial state to the desired state ($H(X) - H(X'|C)$). This is the same as the first concept introduced in [21] for quantifying morphological computation. Note that the choice of controller is relevant to infer any information about the contribution of the open-loop response and hence the morphology itself, even if the controller is open loop. For a closed loop controller, it was also proven that the amount of entropy that can be extracted from a system with initial state X is:

$$\Delta H_{closed} \leq \Delta H_{open}^{max} + I(X;C) \tag{1}$$

Where, $I(X;C) = H(X) - H(X|C)$.

Quantification of morphological complexity is pointless until we can quantify the complexity of a mechanical task itself. From the above works, the closest candidate for the task complexity would be given by:

$$\Delta H_{task} = H(X) - H(X')_{des} \tag{2}$$

Where $H(X')_{des}$ is dependent on the number of independent variables involved and the bounds on the uncertainties that the user specifies. This formulation however does include the states of the agent in it, meaning that that task complexity cannot be quantified without considering the morphology. This is unlike traditional computational problems, whose complexity is independent of the hardware. The complexity of the morphology can be linked to $H(X)$ and the complexity of the controller to $H(C)$ (similar to [23]). In which case, we can then expand the closed-loop controller formulation given in equation 1 to:

$$\Delta H_{closed} \leq \Delta H_{open}^{max} + H(X) + H(C) - H(X,C) \tag{3}$$

From this we can conclude that one way to reduce the complexity of the controller is by increasing $\Delta H_{open}^{max} + H(X)$, which can be done by changing the morphological properties (without changing the cardinality of X), a conclusion which was also found in [23]. In [23], the complexity of the controller was defined as the variance of the control signal (the premise being that a linear controller is the simplest), which for a one-dimensional input is closely related to the entropy. If the cardinality of the state is changed (for example by having additional degrees of freedom), the complexity of the controller is arbitrarily changed as the complexity of the task also changes in the process.

3. Conclusion

Unlike traditional computational problems, it seems that the complexity of control tasks cannot be studied in isolation from the agent(morphology) itself. Hence, hard problems in control are contingent on the robot. For a given task and morphology, the minimum controller complexity (this problem was addressed in [24]) and the optimal open-loop response can be estimated, the latter being the best estimate of morphological computation. Quantification of morphological complexity can be useful to check the *efficiency* of a design, i.e a design whose $(\Delta H^{max}_{open}/\Delta H^{max}_{closed})$ is closer to one would require the least sensory information to perform a task. They would be more robust to sensory noise, however, more sensitive to control output noise [25]. Note that this reduction in control complexity due to the appropriate design of the robot cannot be likened to a morphological complexity-control trade-off. It is possible that for a given task, a simple design with rigid components might provide the optimal open-loop efficiency. Estimation of entropy measures rely on numerical techniques. This makes the process of evaluating a design with its associated controller difficult. It is an open challenge to investigate other metrics for control complexity measures that can be expressed in a closed form, at least with certain simplifying assumptions. A future direction of interest is to look at well known robotic designs and control tasks, evaluating their corresponding task complexity measures, morphological computation metrics and controller complexity to better understand the role of morphology in control and its relation to traditional computational theory.

3.1. Acknowledgments

This work was supported by the SHERO project, a Future and Emerging Technologies (FET) program of the European Commission (grant agreement ID 828818).

References

[1] Pfeifer R, Iida F and Gómez G 2006 *International Congress Series* vol 1291 (Elsevier) pp 22–29
[2] Laschi C and Cianchetti M 2014 *Frontiers in bioengineering and biotechnology* **2** 3
[3] Stepney S 2012 *Handbook of natural computing* **2**
[4] Swade D D 1993 *Scientific American* **268** 86–91
[5] Yasuda H, Buskohl P R, Gillman A, Murphey T D, Stepney S, Vaia R A and Raney J R 2021 *Nature* **598** 39–48
[6] Sauder J, Hilgemann E, Johnson M, Parness A, Hall J, Kawata J and Stack K 2017
[7] Guo N, Huang Y, Mai T, Patil S, Cao C, Seok M, Sethumadhavan S and Tsividis Y 2016 *IEEE Journal of Solid-State Circuits* **51** 1514–1524
[8] Almeida V R, Barrios C A, Panepucci R R and Lipson M 2004 *Nature* **431** 1081–1084
[9] Wang C Y, Chen C W, Jau H C, Li C C, Cheng C Y, Wang C T, Leng S E, Khoo I C and Lin T H 2016 *Scientific reports* **6** 1–7
[10] Church G M, Gao Y and Kosuri S 2012 *Science* **337** 1628–1628
[11] Adleman L M 1994 *Science* **266** 1021–1024
[12] Lovinger D M 2008 *Alcohol research & health*
[13] Ion A, Wall L, Kovacs R and Baudisch P 2017 *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* pp 977–988
[14] Song Y, Panas R M, Chizari S, Shaw L A, Jackson J A, Hopkins J B and Pascall A J 2019 *Nature communications* **10** 1–6
[15] Yasuda H, Tachi T, Lee M and Yang J 2017 *Nature communications* **8** 1–7
[16] Halg B 1990 *IEEE Transactions on Electron Devices* **37** 2230–2236
[17] Nakajima K, Hauser H, Li T and Pfeifer R 2015 *Scientific reports* **5** 1–11
[18] Blum M 1967 *Journal of the ACM (JACM)* **14** 322–336
[19] Pfeifer R and Scheier C 2001 *Understanding intelligence* (MIT press)
[20] Brown E, Rodenberg N, Amend J, Mozeika A, Steltz E, Zakin M R, Lipson H and Jaeger H M 2010 *Proceedings of the National Academy of Sciences* **107** 18809–18814

[21] Zahedi K and Ay N 2013 *Entropy* **15** 1887–1915
[22] Touchette H and Lloyd S 2004 *Physica A: Statistical Mechanics and its Applications* **331** 140–172
[23] Rückert E A and Neumann G 2013 *Artificial Life* **19** 115–131
[24] Montúfar G, Ghazi-Zahedi K and Ay N 2015 *PLoS computational biology* **11** e1004427
[25] Åström K J 2002 *Department of Mechanical and Environmental Engineering University of California Santa Barbara* **333**