Time-Continuous Turing Machines

Lars Š. Laichter

January 2019

1 Introduction

The Church–Turing thesis, with its adjacent construct of the Turing Machine, has constituted the dominant account of computation and the foundation for the Computational Hypothesis in cognitive science [1]. However, in his proposition of the Dynamical Hypothesis, Tim Van Gelder claims the Computational Hypothesis to be irreconcilable with the Dynamical Hypothesis [2]. The Dynamical Hypothesis proposes that dynamical systems are better suited to model cognition than the Turing machine. In the following paper, I challenge this proposition by offering an alternative way of reconciling the Computationalist and Dynamicist divide by altering the original construct of the Turing Machine to better suit the Dynamical Hypothesis. The main objection to the Turing model, which I propose in this paper, is the lack of a continuous temporal dimension. In addition, I use an example of the super-sunflower by Brian Cantwell Smith [3] to propose an account of situated computation better aligned with the Dynamical Hypothesis. I conclude, given the proposed amendments, there does not exist a necessary incompatibility, however, more still must be done to make the Turing Machine an acceptable model to the Dynamical hypothesis.

2 Definitional debate

The question of a proper definition of computation is particularly characterising the divide between Dynamicism and Computationalism. This is not to challenge the success of the Turing model of computation. The Turing Machine has served well in formalizing the notion of "effective computability", thereby answering Turing's question: what can a mechanism do in theory. By establishing a set of computable functions, the notion of "effective computability" underwrites recursion theory, complexity theory, and the 'official' theory of computation. Nevertheless, this conception of computation is subject to critique. I have grounded my argument to a large extent in the work of Brian Cantwell Smith who challenges the idea that there is a 'comprehensive' theory of computation [4]. He proposes three criteria for a potential theory of computation, the *empirical, conceptual* and *cognitive*. The *empirical* criterion requires a theory of computing to explain the full range of computational practice. The

conceptual expects the theory of computation to provide an understanding of what computation is, where it comes from, and what properties are sufficient to consider a system a computational system. And finally, the *cognitive* criterion is that the notion of computation should be clear enough for us to understand what the claim "cognition is computation" means, so that we might be in a position to determine if it is true or false. It is precisely the cognitive criterion which is most relevant to the Dynamical Hypothesis, as if we alter our model of computation, we might also be in a better position to answer the question whether cognition is indeed computation.

3 The dynamical metaphor

In the core of the Dynamical hypothesis, as originally proposed by Tim Van Gelder [5], stands the metaphor of the Watt governor. This metaphor is meant to highlight the salient differences between the computational and dynamical hypothesis. Originally, the Watt governor was designed by James Watt as a way to translate oscillating action of the steam piston into the rotating motion of a flywheel in order to provide a reliable, smooth, and uniform motion for a rotational engine. Its function could be potentially accomplished both by the traditional computational approach, where the governor would be broken down to a set of algorithms, or by the dynamic version of the governor. Of course, the original Watt governor did not apply anything from modern computation. Instead it relied on a mechanical apparatus that continuously regulated steam coming into the engine. Van Gelder [2] argues that the continuous mechanical version of the governor, describable through the theory of dynamical systems, is inherently different from the Turing machine in its capacity to represent cognition.The salient properties are such that: (1) states are the medium of change and have little intrinsic interest; (2) it emphasizes a temporal change of states instead of the particular characteristics of states; (3) it can only be comprehended in relation to other systems, rather than in isolation; (4) its process is primarily viewed as ongoing. Although Van Gelder presents the dynamical hypothesis in opposition to the computational hypothesis, I argue that the views are complementary and together allow for an overall better account of computation, and thus cognition.

4 Rationale

The following proposition of the Time-Continuous Turing Machine is primarily means of reconciling the divide between the Computationalist and Dynamical Hypothesis. However, what would a Turing machine with time-continuous properties similar to those of the Watt governor look like? It is important to note that the Turing Machine does not serve as an exact construct for computational devices, but rather as a central metaphor that amends the particular concept of computation to examination. As Smith [4] points out, the metaphors that

we use are products our conception of computation influenced by our ontological assumptions, methodological commitments, and social and historical biases. The primary two properties, identified as salient, and amended in the following construct of the time-continuous Turing Machine, are **(1)** a temporal dimension; and **(2)** the capacity for participation. These two properties combined later allow for non-effective coupling.

5 Definitions

The structure of the proposed machine consists of four elements: time, tape, table, and the state of the machine. The first element, t, represents time. The position of the machine on the tape is defined as $\phi(t)$. The tape itself is signified as O and is described by the function $O(\omega)$, where ω is an angle and $\phi(t) = \omega$. In contrast to the infinite tape of traditional Turing Machines, the tape in my time-continuous machine is circular. The possible states of the machine can be expressed as a table of states $\{g\}$, where each state has a corresponding configuration and behaviour.

As in the case of traditional Turing Machines, we have a head that moves alongside the tape. For the sake of simplicity, the basic version of this machine moves in a constant circular motion. In the given position $\phi(t)$, which gives us ω, the machine can read a value which we have defined as $O(\omega) \in \langle 0, 1.0\rangle$. That is, the tape is defined as a function that maps angles to real numbers between 0 and 1. This value together with the current state determines in which one of the states $\{g\}$ the machine will find itself. The machine also contains a function $w(t)$, which makes the machine not do any reading or writing, while it continues moving along the tape. The variable W corresponds to the amount of time that the machine is going to wait until it proceeds to the next state.

The content of the tape of the machine O is a continuous function $O(\omega)$. In the most basic version of the machine, it is composed of a set of modifications that can be expressed as a composite linear function. The printing is executed with the function $P(x)$ where $x \in \langle 0, 1.0\rangle$. When the machine prints, such modifications to the tape consist of the *printed value x, location ω, and a modified range r*. It can be written down as $m_x = [x, \omega, r]$. Therefore, we can define the value of the tape as $O = m_1 + m_2 + \ldots + m_n$.

6 An example of a computing machine

The transition tables are defined similarly to Turing's format. This is an example of a machine that prints the sequence 101010... with the starting interval of $W = \frac{1}{2}$, which corresponds to a half of a revolution around the tape. The machine starts at state $g = b$, on a blank tape $O(\omega) = 0$.

	Configuration		*Behaviour*
g	$O(\omega)$	*operations*	*final g*
b	0	P(1.0),w(W)	**b**
	$(0\ ,\ 1.0\rangle$	w(W/2)	**c**
c	0		**b**
	$(0\ ,\ 1.0\rangle$	W=W/2, w(W)	**c**

7 The super-sunflower example

To exemplify the capacity for a temporal non-effective coupling I use an example of the super-sunflower by Smith [3]. Ordinary sunflowers are effectively coupled with the sun. The super-sunflowers are special because they can track something to which they are non-effectively coupled. For example, if the sun sets or goes behind a cloud, the super-sunflower can maintain some coordination with the sun, in such a way that when the sun reappears in some other location, it can still get most of its light and flourish.

The super-sunflower extends Turing Machine's capacity to achieve temporal participation with another system in a continuous-manner. This is primarily because in the originally conceived version of the Turing Machine the time for an operation is defined as constant. On the other hand, the kind of non-effective coupling exhibited by the super-sunflower is easy to capture in the continuous Turing machine framework just introduced.

The example of a super-sunflower can be simplified to two circular tapes, where one represents the sun and the other represents the desired super-sunflower. We call the sun S and the super-sunflower is R. We also introduce the functions $\alpha(S)$ and $\delta(R)$, which stand for the current speed in which the machines are moving. In the case of S, the position of the head $S(\omega)$ represents the position of the sun. To capture the fact that the sunflower cannot directly track the sun at night, we can designate daytime to be the period when $S(\omega)$ is between 0 and π. Thus, R does not have access to $S(\omega)$ if $S(\omega)$ is not in the range π to 2π. In such a case we can say that $\nexists\alpha(S)$ for R. In other words, R does not have access to the speed of S. Both can be initially moving in an arbitrary speed. The machine R can then be defined as:

	Configuration	*Behaviour*	
g	$R(\omega)$	*operations*	*final g*
b	$\exists\alpha(S), \alpha(S) > \delta(R)$	δ(R)+0.1, w(W)	**b**
	$\exists\alpha(S), \alpha(S) < \delta(R)$	δ(R)-0.1, w(W)	**b**
	$\exists\alpha(S), \alpha(S) = \delta(R)$	δ(R)=δ(R),w(W)	**b**
	$\nexists\alpha(S)$	δ(R)=δ(R),w(W)	**b**

Given this definition of the super-sunflower, the machine R would approximate the speed of S based on the size of the amount added or subtracted if the speed of S is smaller or larger than its own. The time it would take also depends on the value of the waiting function. Both the continuously temporal elements

of the R and participation with S would be challenging for a traditional Turing machine.

8 Discussion

Authors, such as Van Gelder, Piccinini, and others have suggested that a continuity is a feature of cognition that fundamentally distinguishes it from traditional Turing Machines [2][6].

Eliasmith argues that the original Turing Machine model is not useful when accounting for cognition and we must thus turn to other computational architectures [7]. However, he recognises the lack of hybrid theories between Turing account, the dynamical account and other conceptions of computation. Although he suggests that hybrid descriptions might lack unification, he also expresses the need to find means of moving between different levels of description in the case of cognition.

Historically, there have been other Turing machine models. For examples. Neary and Woods have explored the possibility of the circular-tape Turing Machines [8]. Similarly, Qu et al. in their recent paper proposed a model for a parallel Turing machine [9] and Chen et al. have written on continuous Turing machines [10].

Currently, as far as we can see, most if not all dynamical systems of practical relevance to cognitive science are effectively computable. Thus, it seems that one way of making the Turing model acceptable to proponents of the Dynamical Hypothesis is to make the model itself more dynamical and resembling cognition. The proposed Time-Continuous Turing machine has attempted to do just that.

9 Conclusions

In conclusion, I have proposed a version of a time-continuous Turing machine that addresses some of the discussed shortfalls of the traditional Turing model. In particular, the lack of a continuous temporal dimension and capacity for participation, which have been appealed to by the Dynamicist. Although the viability of the proposed framework is contentious on a further proof, I hope to have at minimum outlined a possible avenue for a unification of the Computationalist and Dynamicist hypotheses.

References

[1] Alan Turing. On computable numbers. *Proceedings of the London Mathematical Society*, 42:230–265, 1936. ISSN 00246115. doi: 10.2307/2268810.

[2] Tim Van Gelder. The dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences*, 21(5):615–628, 1998. ISSN 0140-525X. doi: 10.1017/S0140525X98001733.

[3] B C Smith. On the Origin of Objects. *MIT Press, Cambridge, MA, USA. Steedman, M*, pages:834–839, 1996. ISSN 1098-6596. doi: 10.1017/CBO9781107415324.004.

[4] Brian Cantwell Smith. The foundations of computing. *Computationalism: new directions*, pages 23–58, 2002.

[5] Tim Van Gelder. What might cognition be, if not computation? *Journal of Philosophy*, 92(2):345–381, 1995. ISSN 13035150. doi: 10.2307/2026571.

[6] Gualtiero Piccinini. Some neural networks compute, others don't. 21(2): 311–321. ISSN 0893-6080. doi: 10.1016/j.neunet.2007.12.010. URL `http://www.sciencedirect.com/science/article/pii/S089360800700250X`.

[7] Chris Eliasmith. How we ought to describe computation in the brain. 41(3): 313–320. ISSN 0039-3681. doi: 10.1016/j.shpsa.2010.07.001. URL `http://www.sciencedirect.com/science/article/pii/S0039368110000336`.

[8] Turlough Neary and Damien Woods. The complexity of small universal turing machines: a survey. URL `http://arxiv.org/abs/1110.2230`.

[9] Peng Qu, Jin Yan, You-Hui Zhang, and Guang R. Gao. Parallel turing machine, a proposal. 32(2):269–285. ISSN 1000-9000, 1860-4749. doi: 10.1007/s11390-017-1721-3. URL `https://link.springer.com/article/10.1007/s11390-017-1721-3`.

[10] Xiaoliang Chen, Wen Song, Zexia Huang, and Mingwei Tang. Continuous turing machine: Real function computability and iteration issues. 8(5): 2405–2416. ISSN 1935-0090, 2325-0399. doi: 10.12785/amis/080536. URL `http://www.naturalspublishing.com/Article.asp?ArtcID=5911`.