Investigating in-vitro neuron cultures as computational reservoir

Peter Aaser

***Abstract*—Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.**

Keywords*—*neurons, reservoir-computing, cyborg

I. Introduction

Since the 50s, the Von-Neumann computer architecture has been ubiquitous in the field of computing. This architecture owes its success to its relative simplicity, allowing quick iterations in tact with doubling transistor counts in accord with moores law. However several issues with the Von-Neumann architecture have become more and more pressing in recent times. Issues such as the Von-Neumann memory bottleneck are becoming increasingly difficult to hide, and its inherently serial nature of computation. Even in the case that moores law will continue, we still face issues with lack of scalability in modern processors. With billions of transistors a top down design process is becoming increasingly difficult and expensive, and a single faulty transistor can cause an entire chip to be useless. These weaknesses necessitates going beyond the Von-Neumann architecture, exploring unconventional computing paradigms. Taking inspiration from nature we look at cellular computing consisting of computational cells which by themselves are unremarkable. In these systems we observe properties that cannot be traced back to a single cell, they only appear when multiple cells are interacting with each other. In nature these systems form without any form of supervision, a property known as *self organization*, with an example being DNA which “describes how to build the system, not what the system will look like” [1]. These *emergent properties* arise in systems built from the bottom up by a set of growth rules rather than a designers intent, a property, which allows a flexibility and robustness that modern processors lack, a prime example being the human brain. The human brain is a vastly parallel computer, eclipsing modern processors in terms of computational capacity, robustness, energy efficiency and complexity. The three first properties make them very appealing subjects to study for unconventional computing, however due to the immense complexity the brain is still enigmatic. In this paper we explore using neuron cultures to create an organic mechanical hybrid organism, a so called *cyborg*.

II. Background

A. The NTNU Cyborg Project

The NTNU cyborg project is a collaboration between several departments at NTNU including the department of biotechnology, computer and information science, engineering cybernetics, neuroscience and more. [**?**] The stated goal for the cyborg project is “to enable communication between living nerve tissue and a robot. The social and interactive cyborg will walk around the campus raising awareness for biotechnology and ICT, bringing NTNU in the forefront of research and creating a platform for interdisciplinary collaborations and teaching.” Currently the department of neuroscience is growing neuron cultures which are to be used as the biological part of the robot. These neuron cultures are not part of a brain, they are fully dissociated, grown in special chambers in-vitro. The robot part of the cyborg has been developed and implemented by the department of engineering cybernetics, and is currently operational, however it has not yet been integrated with the in-vitro neuron cultures. The challenge faced by the cyborg project is the infrastructure for interfacing the neuron cultures and the robot, essentially creating a brain computer interface.

B. Complex Systems

In this paper references are made to computations done by both artificial and real neurons. To make the distinction between these cases clear all computation done by computer simulated approximations of neurons will be prefixed as artificial.

The self-organizing properties that makes cellular computing such as neural networks so successful in nature also make understanding and exploiting them difficult. In contrast with microprocessors, cellular computing describes systems where each component cell is relatively simple, interacting mostly with its neighbors with no global control mechanism. Neuron cultures pose additional difficulties compared to simpler cellular systems, both practical and complexity-wise, as even a single neuron is a vastly complex entity. To approach neuron cultures it is useful to employ models of simple cellular systems that exhibit self organizing and dynamic behavior, known as *complex systems*. Many such systems have been explored, such as recurrent artificial neural networks [2], random boolean networks [3] and cellular automata. In [4] Langton explores the requirements for systems to support emergent computation, where he argues that in order for a system to support emergent computation it must lie in a *critical phase* between order and chaos, drawing parallels to phase transitions in material science. This observation comes from thermodynamics, in which a material may exhibit a *second order phase transition* between a solid and liquid form where the material undergoes a continuous transition in contrast to a first order transition such as melting ice. Using cellular automata as an example, he explores the effect of varying the rules for calculating the next state with a parameter, λ, which describes how likely it is for a cell to enter a *quiescent state*, a state where it will not disturb other cells until it leaves the quiescent state itself. At $\lambda = 0$ all cells enter a quiescent state after one step, representing a fully ordered system, thus at $\lambda = 0$ the system is in the ordered phase. At $\lambda = 1$ there are no rules that leads to a quiescent state, leading to very chaotic systems with very high entropy, representing the chaotic phase. As λ is increased from 0 the cellular systems starts to form intricate structures, increasing in complexity and taking longer to reach a steady state. This behavior peaks at $\lambda \approx 0.5$ which is the most critical phase in this particular system, creating complex self-organizing structures. As λ is increased further the self-organizing structures start to give way to more chaotic and seemingly random behavior, gradually transitioning into the chaotic phase.

It turns out then, that with most cellular systems providing the necessary conditions for computation to occur is a tractable problem, however harnessing and shaping this computation is a whole different matter. In their critical phase systems are unstable but not chaotic which causes them to be highly sensitive to small changes in topology and initial conditions. Attempting to maneuver this fitness-landscape by directly changing the properties of a system would in many ways be like calculating the correct way for a butterfly in china to flap its wings in order to cause a hurricane in New York.

C. Reservoir Computing

Faced with the intractability of designing cellular computation systems in a top down manner, and the unpredictable relation between cause and effect in self organizing systems has made it necessary to approach complex systems in a different manner. One such approach is to treat the system as a *reservoir* [5] which "acts as a complex nonlinear dynamic filter that transforms the input signals using a high-dimensional temporal map, not unlike the operation of an explicit, temporal kernel function."

In reservoir computing there is no designer shaping a system, the systems organize and shape their own behavior, forming systems with complex behavior that can reflect subtle differences in initial conditions and stimuli. From the perspective of reservoir computing using a neural culture is reduced to learning the correlation between input and output from the reservoir, treating the internals as a black box. 1 shows a typical RC (reservoir computing) system with three inputs and two outputs. The inputs are processed in a simple feed-forward neural network before perturbing the reservoir in some way. Similarly the state of the reservoir is being processed by an output layer before leaving the RC system. In the figure the input and output processing is done by feed forward neural networks, but we note that this is only one of many possible filters. Inputs 1, 2 and 3 are snapshots of the current state of the problem we attempt to solve with reservoir computing. Since our filters have no state, at least not beyond some time horizon we see that the history of the system must in some way be encoded in the reservoir in order for the RC system to solve problems in scope wider than the limited amount of state that may be contained in the filters.

D. Neuron Computing

Neurons are cells which communicate with each others using electro-chemical signals in vast interconnected networks. These networks communicate in a complex interplay between neurotransmitters and voltage spikes, and their communication prompt structural changes in the network, deeply intertwining the emergent behavior and self organizing properties on neurons. There is a high degree of correlation between the electrical, chemical signals in these neural network, thus a necessary simplification of considering only the electrical signals when interacting with neural networks can be made at a small cost of information loss. These electrical signals, known as action potentials, are caused by polarity differentials between the ions inside and the ions outside of the cell membrane. A neuron maintains this potential difference by transporting ions of opposite polarity through the cell membrane, essentially preparing a spike of electrical activity. This spike is triggered as the neuron receives electrical stimuli via a network of electrical receptors called dendrites When excited, a neuron which has built up a sufficient polarity difference will trigger a polarity shift. This polarity shift cascades along the axon, a long tendril that can extend to reach far away neurons, forming branches to reach multiple neurons.

III. A Hybrid Neuro-Digital Approach

1) Concept: The main challenge creating a cyborg is harnessing and applying the computational power of neurons to enhance the capability of the robot, showing that the neurons are actually doing something useful. As previously discussed the function of neural networks are vastly complex, attempting to fully understand the complex interplay of chemical and electrical signals is intractable. Therefore, in order to interface with the neural network we employ the mindset of reservoir

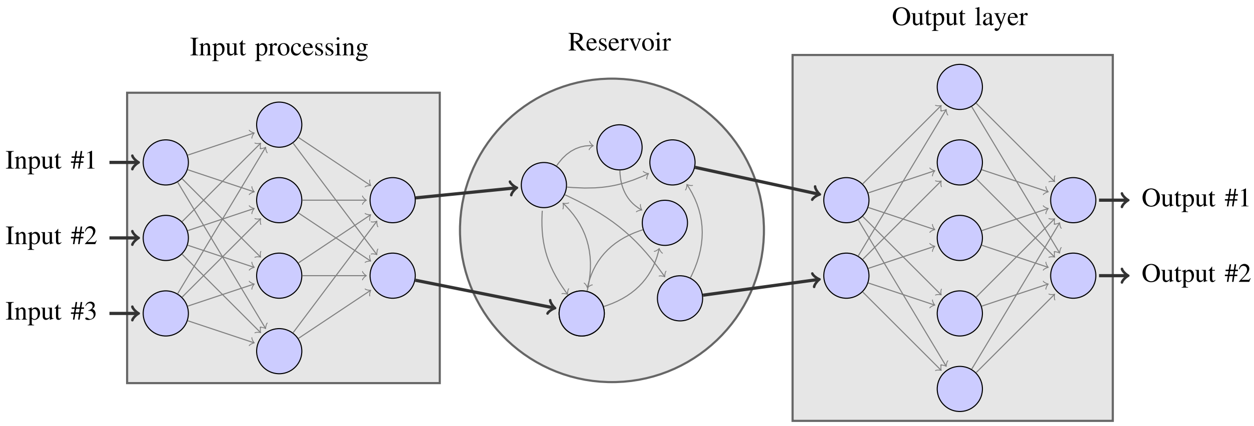

Fig. 1. A reservoir comput-thingy

computing. From this perspective we can reduce the complexity of the task significantly. The first benefit is that we can choose to focus on only electrical signals, which vastly reduces the search space at a minimal information loss cost. Secondly we have now avoided having to explicitly train the neurons towards performing their task, we now view them as a reservoir for dynamics which responds to stimuli, leaving the task of translating this response into useful actions. In 3 this approach is visualized: A neural network is grown in a *micro electrode array* which allows interfacing with a neuron culture by measuring and inducing voltages on an array of electrodes which the neurons grow around. This partial view of the complete system is interpreted by an artificial neural network which outputs commands to a robot. Similarly feedback is provided to the neural network from the sensors of the robot which is translated into electrical stimuli requests by a feedback processor.

2) Platform: Providing an interface between neurons and a computer is an important first step, however it is highly impractical to move the neural cultures outside of the laboratory. The practical solution is quite thought provoking[1], by using a TCP/IP network protocol the neuron culture may be interfaced with any robot or simulator without leaving the laboratory. Similar network architectures have been implemented, in [6] a neuron culture is used to control a simple wall avoiding robot, ## something about the shortcomings of this paper ##.

3) Growing NIV:

4) A First Test:

IV. Results

Here's where we add the results

V. A Hardware/Software RC Platform

A. MEA2100

The hardware used to interface with neuron cultures for the cyborg is an MEA2100 system purchased from multichannel systems. The MEA2100 system is built to conduct experiments on in-vitro cell cultures, with the main focus being on neurons. The principal components of the MEA2100 systems are:

1) Micro electrode array: To experiment on the properties of cells or other electrically active subjects the micro electrode array (MEA) is used. As the name implies the MEA is equipped with an array of electrodes able to both measure the electrical properties of the experiment subjects, as well as applying outside stimuli, acting in a sense as output and input for the subject. 4 shows an empty MEA, **??** shows an MEA from st.olavs with a live neuron culture.

2) Headstage: The electrodes of the MEAs are measured and stimulated by the headstage which contains the necessary high precision electronics needed for microvolt range readings. 6 shows the same type of headstage used in this paper along with an MEA.

3) Interface board: The interface board connects to up to two head-stages and is responsible for interfacing with the data acquisition computer, as well as auxiliary equipment such as temperature controls. The interface board has two modes of operation. In the first mode the interface board processes and filters data from up to two headstages as shown in 8 which can then be acquired on a normal computer connected via USB. In the second mode of operation a Texas instruments TMS320C6454 digital signal processor is activated which can then be interfaced with using the secondary USB port as shown in **??**

Designing and implementing the architecture has revealed weaknesses with previous

VI. Conclusion

We conclude that this was an excellent idea but it would be more fruitful to investigate distributed webscale apps.

VII. Further Work

For instance investigating growth rules for neurons in chaos or orderly environments to investigate if they trend towards a critical phase.

[1]For brevity the philosophical implications is left as an exercise to the reader.

Acknowledgment

The authors would like to thank...
Sandoz methylphenidate 54mg, couldn't have done it without you.

References

[1] G. Tufte, "From Evo to EvoDevo: Mapping and Adaptation in Artificial Development,"

[2] N. Bertschinger and T. Natschlger, "Real-time computation at the edge of chaos in recurrent neural networks," vol. 16, no. 7, pp. 1413–1436.

[3] C. Gershenson, "Introduction to Random Boolean Networks,"

[4] C. G. Langton, "Computation at the edge of chaos: Phase transitions and emergent computation," vol. 42, no. 1, pp. 12–37.

[5] B. Schrauwen, D. Verstraeten, and J. M. V. Campenhout, "An overview of reservoir computing: Theory, applications and implementations," in *ResearchGate*, pp. 471–482.

[6] Y. Li, R. Sun, B. Zhang, Y. Wang, and H. Li, "Application of Hierarchical Dissociated Neural Network in Closed-Loop Hybrid System Integrating Biological and Mechanical Intelligence," vol. 10, no. 5, p. e0127452.

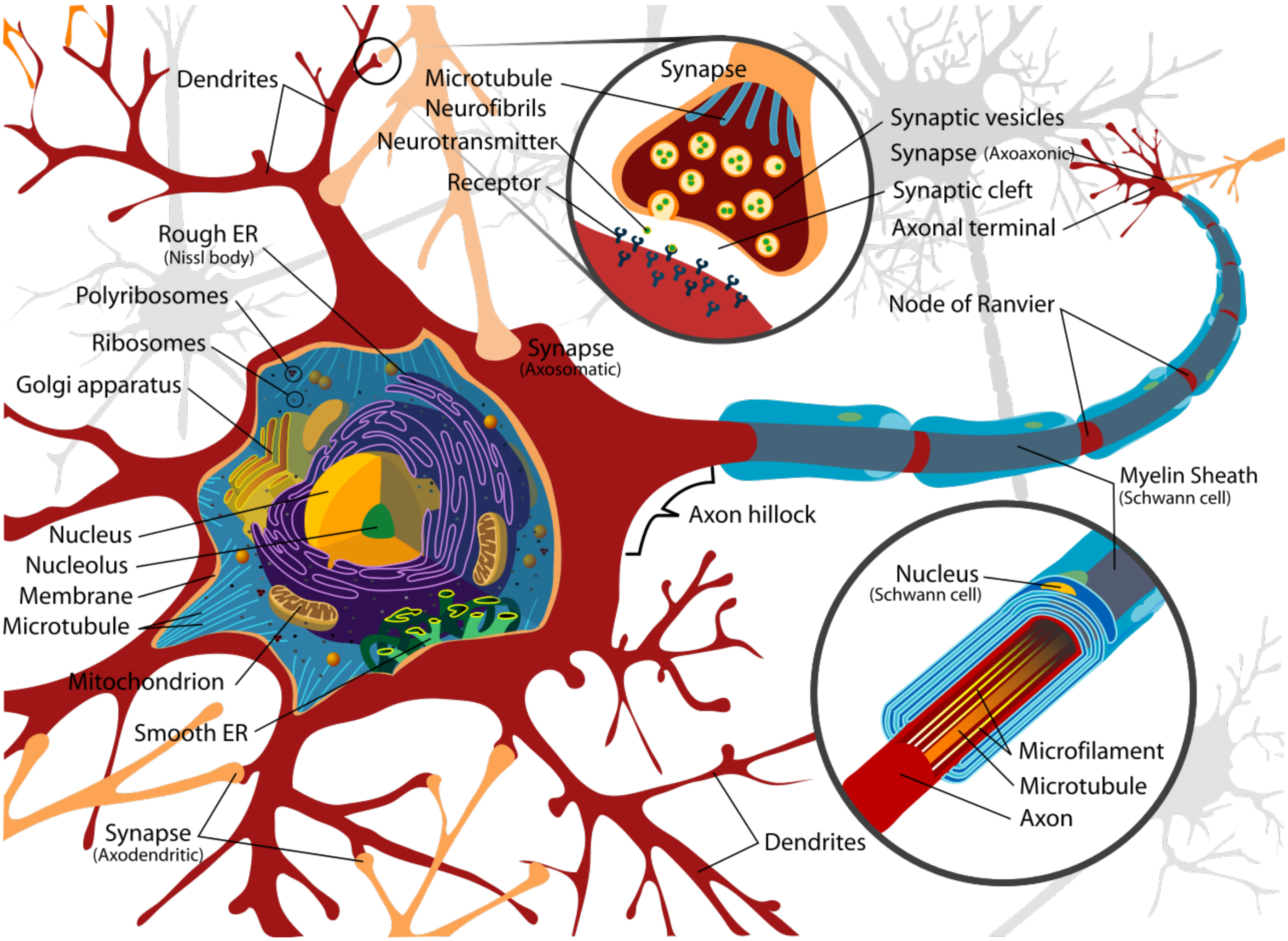

Fig. 2. a neuron

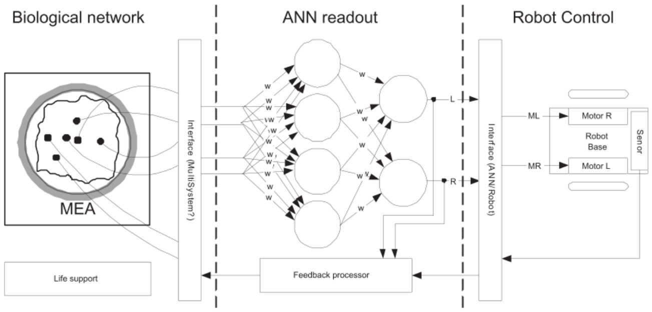

Fig. 3. The gist of it..

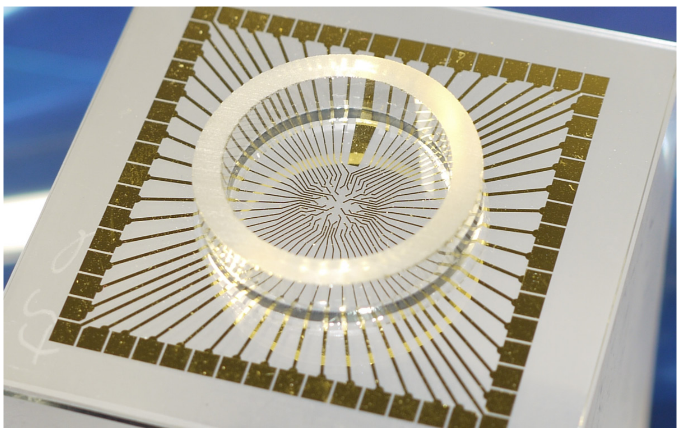

Fig. 4. A generic MEA

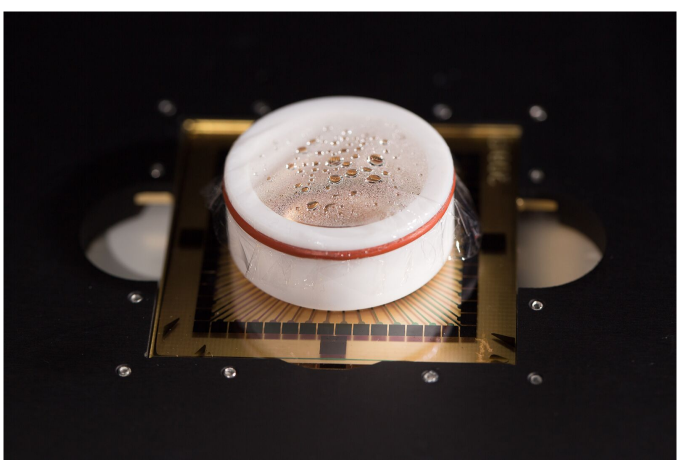

Fig. 5. A MEA with a live culture, photographed by Kai

Fig. 6. The headstage

Fig. 7. The MCS interface board

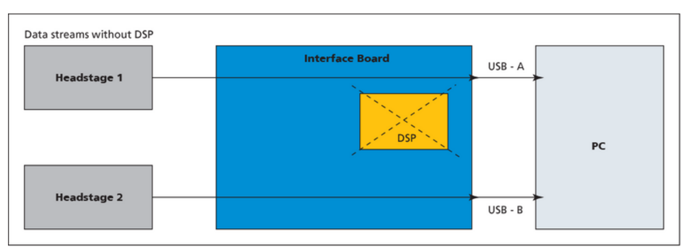

Fig. 8. Casual mode

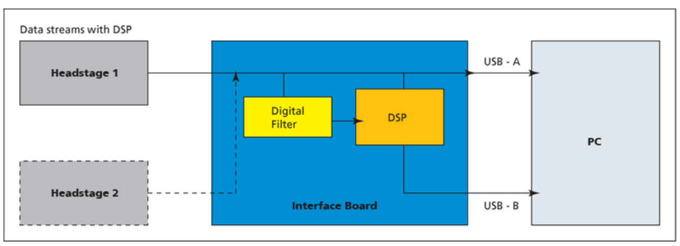

Fig. 9. DSP active