# 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*—*memes, reddit, unfunny, low-energy***

## I. Introduction

Reservoir Computing is a novel approach to machine learning, using reservoirs containing dynamic complex systems as computational elements. [1] Many different systems have been used as reservoirs, ranging from virtual reservoirs such as random boolean networks and recurrent neural nets, to real physical systems such as nano-carbon tubes. Building on previous work we investigate using in-vitro neuron cultures as reservoirs, describing the necessary infrastructure for performing such experiments with live neurons. The primary concern of this paper will be the architecture necessary for embodiment of blah blah blah

## II. Background

### A. Reservoir computing

A big disconnect between machine learning and how we perceive human reasoning is how to solve and encode problems of temporal nature. While feed forward structures which can only consider their current input can be extended to reduce temporal problems into structural ones by adding delays [1], a more natural approach is to use recurrent structures that change their internal dynamics in response to input. A widely studied example of this approach is the recurrent neural network, however the increase in expressiveness makes these structures very hard to train[2]. Another example are random boolean networks [3] studied by Kauffman as a model for genetic regulatory networks. Common for these two systems, and many more such as cellular automata [4] or even a bucket of water (?? cite EFPL bucket paper ??)is that they exhibit the properties of *complex systems* [5]. Complex systems are systems that are, in a sense, magic. In order to harness the power of these complex systems it is fruitless to attempt to shape the topology and dynamics of the system towards some specific goal. Instead, the complex systems are used as *reservoirs* which we can interact with using a simple linearly separable input and output processing layer which can be easily trained [1]. (??EFPL paper??) 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.

### B. Neurons

Neurons are vastly complex entities, communicating through complex electric and chemical signals. However, since we are more interested in the emergent properties of neurons in the context of reservoir computing a superficial description suffices. We will only consider a generalized version of the neuron, but in our experiments a plethora of different neurons are used, although they all share the basic similarities described here. The anatomy of a neuron is shown in 8 and can roughly be divided into the following parts:

*1) Soma:* The main body of the neuron. While we will view neurons as simple network nodes it is important to note that the neuron is highly complex, it can blah blah

*2) Dendrites:* To sense its surroundings the neuron is equipped with dendrites. These branching structures act as receivers, propagating electro-chemical stimuli to the cell body. Their reach is only to the immediate vicinity of the cell, they do not form longer connections.

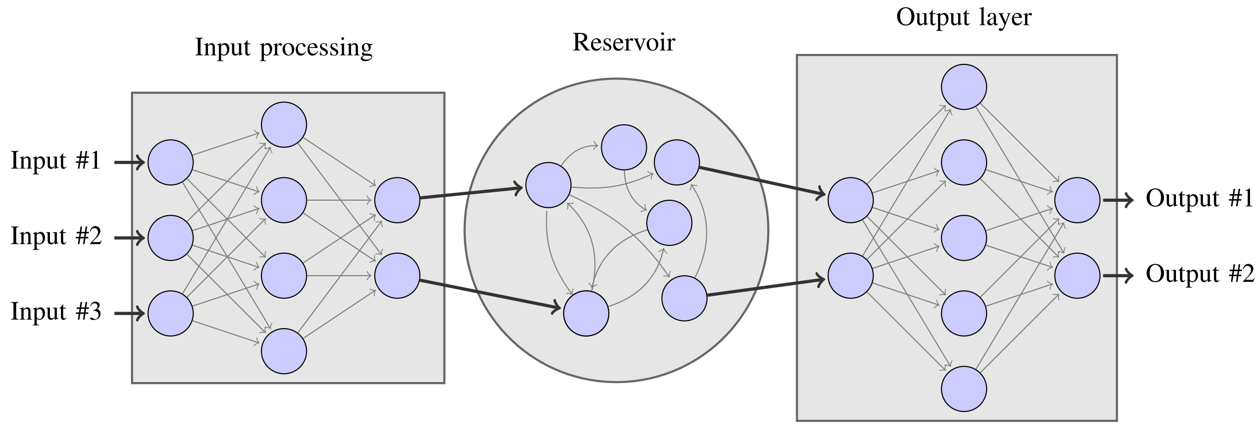


Fig. 1. A reservoir comput-thingy

*3) Axon:* The axon is a long tendril, extending over a meter in the case of the sciatic nerve TO-DO maybe embed link? which transmits information as electrical pulses to other neurons. An axon can branch off and reach multiple neurons, it is not a one to one connection.

### *C. Neuron-cultures as Reservoirs*

Biological neurons have inspired artificial neurons, so BLARGH

### *D. 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."

### *E. MEA2100*

To perform experiments a MEA2100 system has been 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. 2 shows an empty MEA, **??** shows an MEA from st.olavs with a live neuron culture.

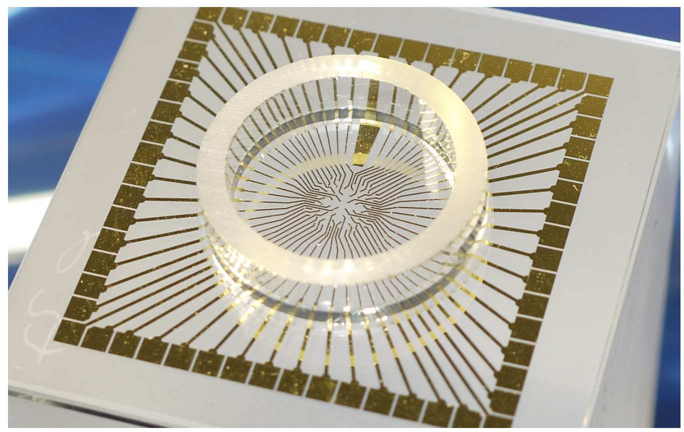

Fig. 2. A generic MEA

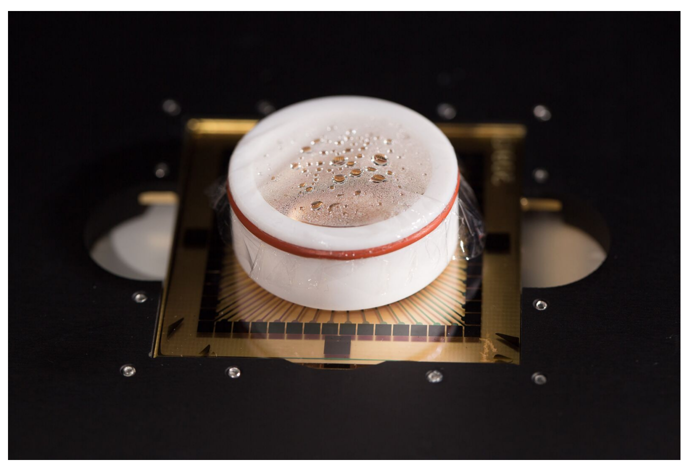

Fig. 3. A MEA with a live culture, photographed by Kai

*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.


Fig. 4. The headstage

4 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


Fig. 5. The MCS interface board

operation. In the first mode the interface board processes and filters data from up to two headstages as shown in 6 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 **??**

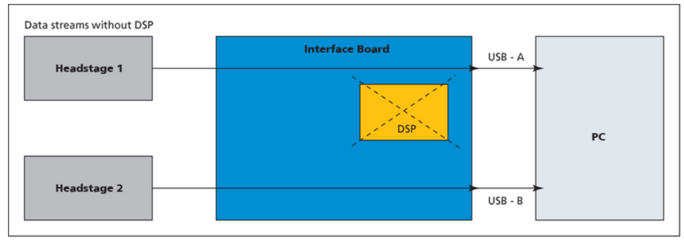


Fig. 6. Casual mode

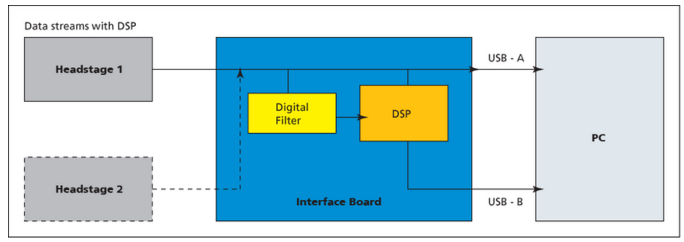


Fig. 7. DSP active

## III. Methodology

TO-DO: differentiate RC-agnostic post processing and spike detection. Should probably be done in background

The architecture described in this paper is a refinement of the neuro-robot architecture used in [6]. In [6] a system for working specifically with MEAs containing dissociated neurons is described with a TCP/IP connection to a slave PC controlling a robot being the main selling point. Our model is a generalization of [6] designed for reservoir computing, where the neuron-culture is just one of many possible reservoirs. It should be noted that our architecture does not require experiments to be posed in the context of reservoir computing. (not too happy with the words here, make them betterer. Also something about the possibility of multiple implementations)

### A. Areas Of Concern

The architecture has been defined by three areas of concern. This modularization facilitates reuse of code ??? some words ??? In this section it is important to note that we differentiate between generic data-processing and task specific processing. The former describes getting spike trains while the latter describes reservoir computing specific processing This is good because it separates RC from other stuff

*1) Data Acquisition and Interfacing:* Data acquisition entails configuring the MEA2100, collecting data from the MEA2100 and setting up stimuli. This means the data acquisition and interfacing software must at least be able to deliver a raw, that is unprocessed data-stream, as well as being able to accept requests for stimuli over TCP/IP. In practice the data acquisition software may also handle processing of acquired data for performance reasons, but it is not a required task.

*2) Data Processing:* This part of the architecture is responsible for processing data in a task specific, i.e non-generic way. The data may be raw waveform data or spike data from the MEA that needs to be processed into commands for an agent, or it may be sensor data from a simulated agent that must be processed into stimulus requests for the MEA. As with the data acquisition and interfacing module, the data processing module communicates over TCP/IP.

*3) Agent Control:* This module implements the actual embodiment of the neuron-culture. The agent control reads the processed data from the data processing modules and issues commands to an agent. It also transmits sensor data back to the data processing module, providing feedback to the neuron culture. The type of the agent is not specified, it can be a fully fledged walkin' talkin' cyborg, or it can be a simple simulated agent.

### B. Implementation

Currently the following software systems have been implemented:

*1) MEAME:* MEAME implements the data acquisition and interfacing requirement of the closed loop system. It is written in C# and interfaces with an API provided by multichannel systems which allows it to interface with the MCS interface board. MEAME also implements the optional DSP which is used to stimulate the neurons thus fulfilling both the requirements.

*2) SHODAN:* SHODAN is a framework for composing reservoir computing experiments written in scala. In contrast to MEAME which is written specifically for interfacing with MEA2100, SHODAN is intended to facilitate general purpose input and output processing for reservoir computing. SHODAN currently hosts a number of tools for composing and reusing data-processing pipelines such as parametrizable feed forward neural nets, serialization and deserialization methods for TCP streams for communicating with the data-acquisition module and the agent control module. Additionally SHODAN hosts a simple simulator intended to stand in for a more full-fledged agent control module. This simple simulator simulates a simple agent with four eyes which reacts to proximity to a wall, and can visualized in a browser in real-time to get that sweet grant $$$

### C. Interfacing With the MEA2100 redux

## IV. Results

Here's where we add the results

## V. Conclusion

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

## VI. Further Work

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

## Acknowledgment

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

## References

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

[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] M. Sipper, "The Emergence of Cellular Computing," vol. 32, no. 7, pp. 18–26.

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

[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.

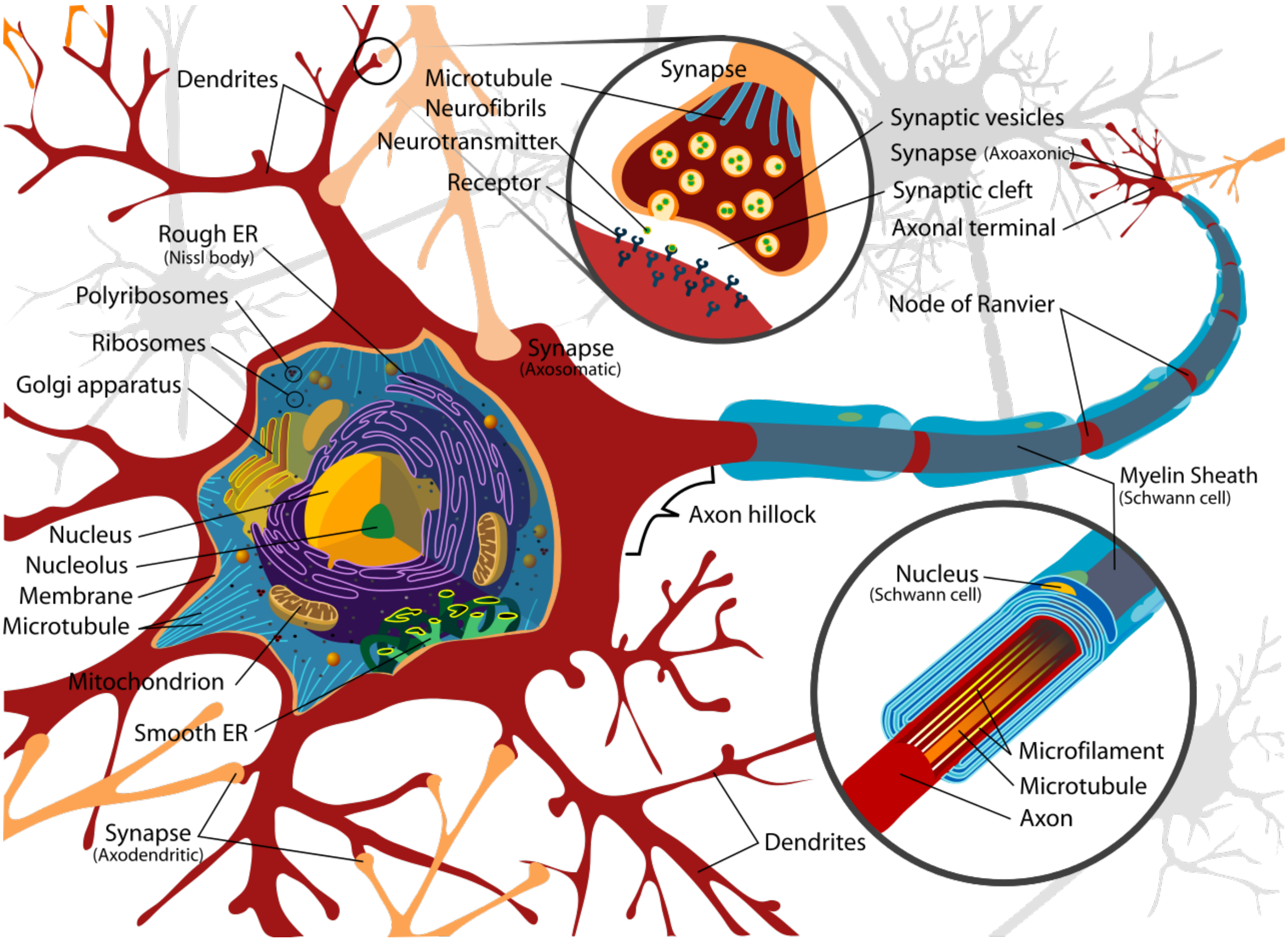


Fig. 8. a neuron