Technical Report

Brain and Simulation

This report focuses on the human brain, more so the cerebellum and how we can formally describe neural networks, and examples of how these practices can be merged to make way for a breakthrough towards the ultimate goal of modelling the human brain.

Ally Malima
1/12/2015

Contents

Introduction

We know that the brain is an amazing type of super computer, with regards to all the millions of different functions it performs every single day of our lives! For this reason scientist all over the world are interested finding out just how interesting it really is and what exactly can be done to better understand this small yet incredibly complex part of our bodies. In this report I will aim to show firstly just how complex the brain really can be by using the cerebellum as a prime example, as it is closely related to my own personal project interests. Then of course once we know what the cerebellum is and does, how we can proceed to describe what we observe on a biological scale formally via use of simulator. All of this leading up to the discussion of the different methods and techniques available and how they measure up to the challenge of modelling a type of super computer that has come ages before it. Interestingly enough we find that what we currently does quite a good job of doing so not only that but it's showing so much promise that it has motivated an EU flagship initiative The Human Brain Project lasting ten years committed to realising "a new "ICT-accelerated" vision for brain research and its applications".

1.1 Brain, Cerebellum and Movement

We know that the brain is an amazing type of super computer, with regards to all the millions of different functions it performs every single day of our lives! Thanks to much in depth research we know the structures and roles played by different areas and parts of the brain.

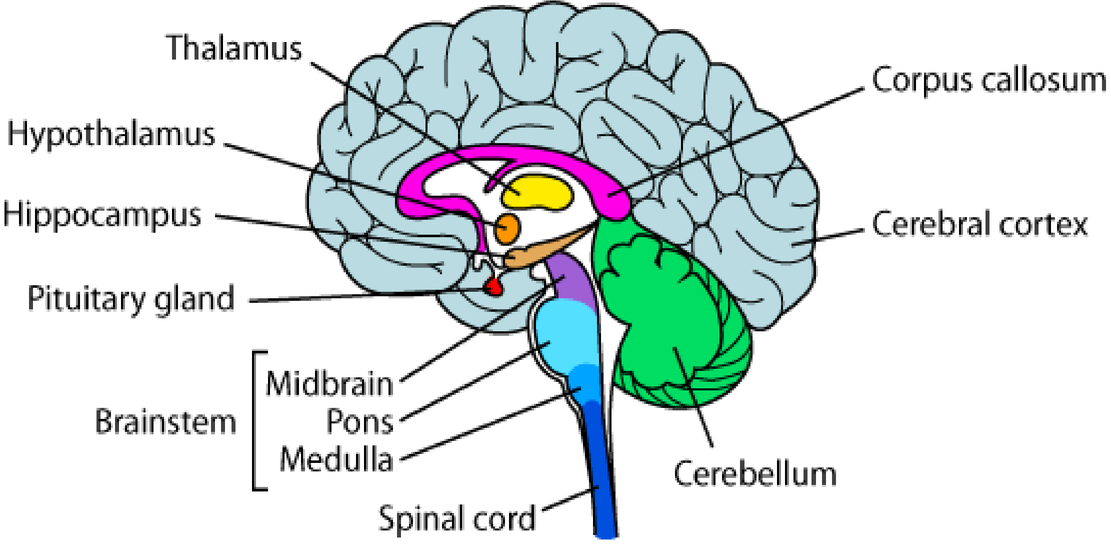

Figure 1 Inside vertical view of Brain and Cerebral Cortex (in text citation here)

Fig. 1 shows an inside view of the brain. Different areas are responsible for different functions. In this report we will focus on the cerebral cortex and even more on the cerebellum.

The cerebral cortex is the largest part of the brain and is responsible for learning, thinking and also control and initiation of voluntary movement of our limbs.
The cerebral cortex together with the cerebellum makes up the most part of the Cerebrum (Brain). The cerebral cortex is responsible for many processes in the cerebrum; learning, thinking etc. but undoubtedly the most important being control and the initiation of motor movement of our limbs i.e. picking up a pen.

The cerebellum is the smaller structure located dorsal of the Pons. From Fig.1 we observe that the cerebellum is significantly smaller than the cerebrum, it occupies roughly 10% of the volume of the brain. Astoundingly, it still contains approximately 50% of the neurons in the central nervous system. It was found that people with damage to the cerebellum or animals with lesions in the cerebellum were not necessarily unable to conduct or initiate motor movement but suffered serious impairments to the coordination and 'smoothness' of these movements.

So, from this we may conclude that the cerebral cortex has a very important role in motor movement as without it there would be no movement! However, the cerebellum arguably plays an equally important role in coordinating our motor movement so that it is well-estimated, and done with precision and accuracy.

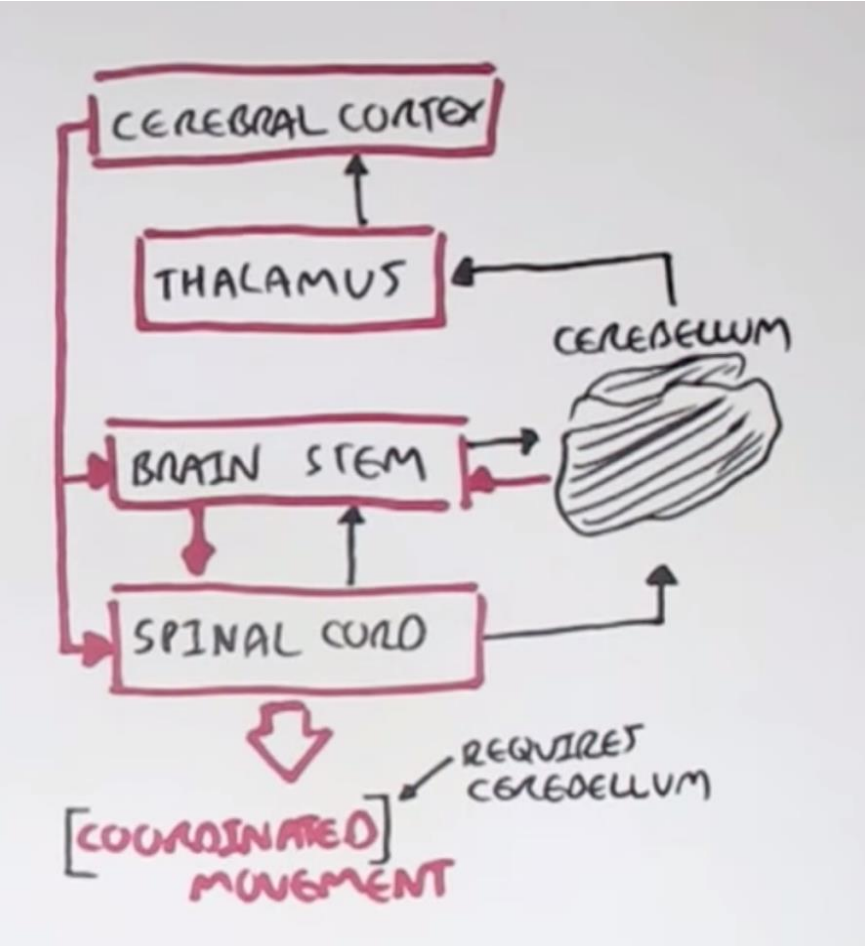

Figure 2 The overall role of the cerebellum (in text citation)

Fig 2 aims to show an overall picture of the role of the cerebellum by highlighting the loop of information sending as a result of the presence of the Cerebellum. Starting from the top of the diagram we see that information is sent from the cerebral cortex to the brain stem and spinal cord this information is basically the plan of action i.e. move your arm. Simultaneously, the spinal cord is sending signals to the brain stem and the cerebellum about the position of limbs. Then, from these two pieces of information from the cerebral cortex and the limbs the cerebellum then precisely and carefully calculates and creates a blueprint of how best the cerebral cortex can implement this movement including balance, force and speed. This blueprint is then sent to the cerebral cortex via the relay nuclei in the thalamus. Meanwhile, the Cerebellum also sends a signal to the brain stem and spinal cord to assist in this coordination process. Therefore, we reiterate that although we do not need the cerebellum to make a movement, it still roves to be vital because it is required for the *coordination* movement.

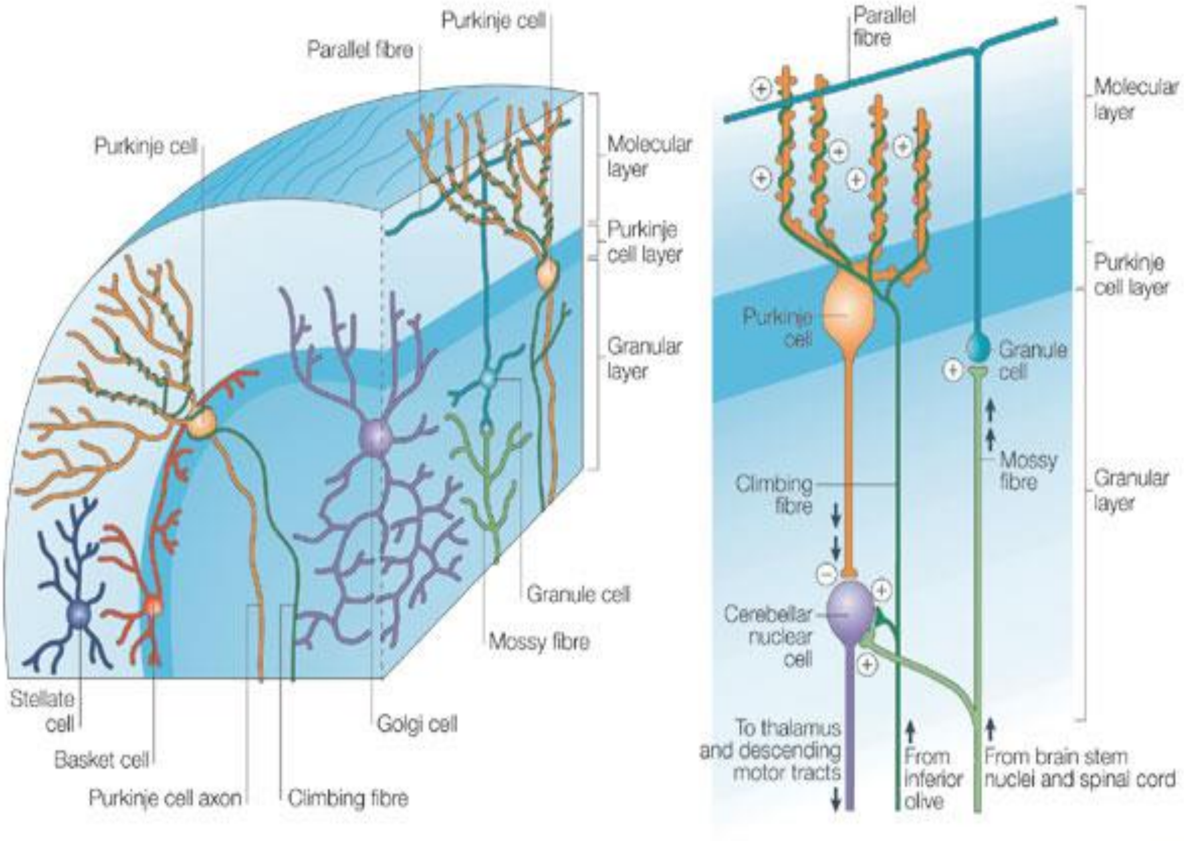

Figure 3 Cellular anatomy and circuitry of the cerebellum (in text citation)

Fig.3 shows cellular anatomy of the cerebellum and of the circuitry.

The functional organization of the cerebellum can be functionally distinguished into 3 different components; Vestibulocerebellum, Spinocerebellum and the Cerebrocerebellum.

- The Vestibulocerebellum is involved with the management of balance,
- The Spinocerebellum or (vermis) is important in managing your stance and gait (ability to walk)
- The Cerebrocerebellum is involved in the management of smooth and precise control.

In order to appreciate how the cerebellum can accomplish these 3 things we have to appreciate how the components of the cerebellum are connected.
The cellular anatomy of the cerebellum separates into three layers. The Molecular layer, the Purkinje layer and the Granular layer.

The Molecular layer is closest to the surface of the cerebellum and contains 'information receiving' parts of the Purkinje cells, their dendrites, 'information sending' portion of the granular cells, their axons (also known as Parallel or T-Fibers) and lastly several interneurons which are inhibitory neurons responsible for modulating all the activity.
The Purkinje layer contains the Purkinje somas (or cell bodies) which send their dendrites up to the molecular layer, in addition there are also astrocytes (also known as Bergmann cells), which help in the maintenance of the Purkinje cells.
Lastly, The Granular layer contains the granular somas which send their axons known as T-fibers up into the molecular layer, which also contains Golgi cells which are specialized interneurons also inhibitory. The important point to note is that the purkinje dendrites meet granular axons which then synapse to produce simple low frequency spikes (signals).

So from this breakdown of the cerebellum we see that the brain is great, however at the same time very large scale and vast in complexity. So in order to model this all we have to find an acceptable functional level. After further research into the cerebellum itself, we found that the cerebellum could be divided further into functional zones consisting of smaller microzones, which are the minimum functional effective units of the cerebellum.

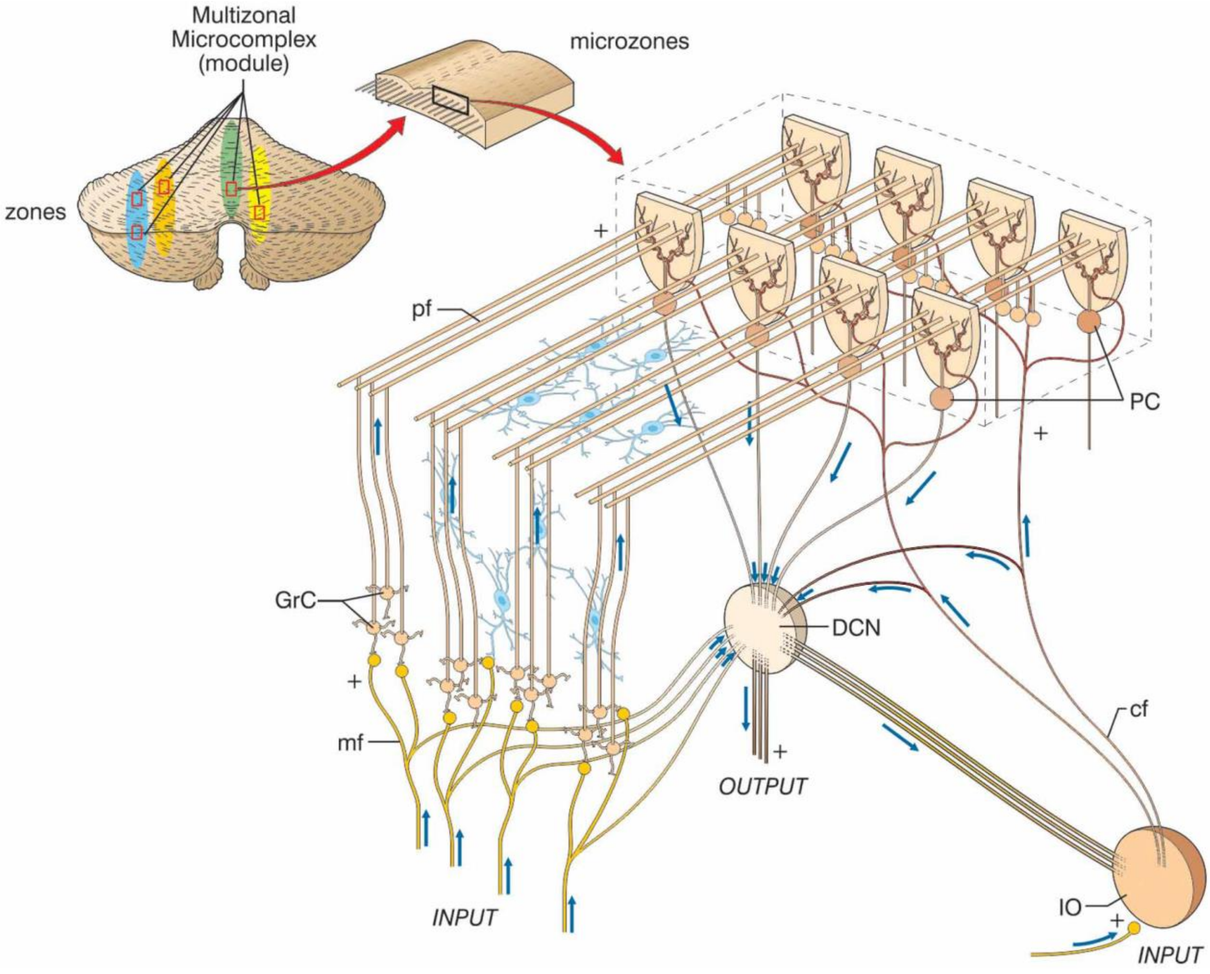

Figure 4 Cerebellum at the microzonal level (in text citation)

Fig.4 shows the structure of a microzone. From this diagram it is possible to describe briefly but concisely how the inputs are translated into an output that projects to the thalamus and then to the motor cortex.

We have 2 inputs into the cerebellum, the first input is the sensory nuclei (from the brainstem and spinal cord) and the inferior olive from the medulla oblongata.
Mossy fibres from the sensory nuclei synapse on the granule cells at this region called the 'rosettes' these granule cells' axons become horizontal called parallel fibres which innervate the Purkinje cells via interactions with their dendrites this synapse of parallel fibers with the Purkinje cell dendrites produces simple spikes at low frequency.
The second input is the inferior olive from the medulla via the Climbing fibers which synapse on the Purkinje cell dendrites and the soma to produce complex and high frequency spikes. Those spikes integrate and the output will be synapsing with the deep cerebellar nuclei and is an inhibitorty synapse which then projects to the thalamus which in the end projects to the cerebral cortex that initiates the movement!

To summarise the Cerebellum receives two major inputs, one from the sensory system (spinal cord) and the other from the other parts of the brain. It then integrates these 2 inputs and fine tunes our motor activity. Hence, the effect of cerebellum lesions is not paralysis but rather a disorder of fine movement, such as balance and adjustment of motion.

1.2 Describing Neural Networks

This chapter aims to explain how we formally describe neural networks. From the previous chapter the depth and structure in which the neural pathways are exhibited really gives us a huge need for something to model or describe these neural neural networks. Thanks to computational neuroscience we have a vast range of software's to simulate neural networks. In this chapter I will aim to compare PyNN and Neuro ML.

However, it is important to recognise that there are both pros and cons to having a vast range of simulators. The cons are that it can be harder to use a sample piece of work from someone else and import models of systems onto your personal simulator as individual simulators use their own programming languages. The pros are that because this simulator diversity means that different simulators are better than others or different functions therefore you can choose a specific simulators if you want to design a specific model and also results from a particular simulator can then be cross-checked with those of the others to find the optimal simulator for e.g.

Therefore, a simulator independent language would solve the issue posed by the variety of simulators, It turns out there is the PyNN interface which uses Python as a common programming language so the idea is that you can write a simulation script in Python and simply run it without editing it as it is supported by all the simulators (NEST, NEURON, Brian etc.)

As mentioned before this is very useful because it allows people easier access to example code build on it whilst also making it simpler to evaluate results from different simulators. Lastly, it is an open source software.

NeuroML is another approach to model the underlying complex connectivity of the brain. As mentioned before while simulators NEURON, NEST, Brian are useful in modelling this, they are not interchangeable between simulators and this can make it harder to assess the validity of the model amongst other simulators. This is what sparked the use of NeuroML, as ‘a neuronal model description language based on XML (Extensible Markup Language)’.

The benefit of this is that models can be defined in a form independent of a specific simulator i.e. they can be implemented in many simulators and stored in a standard format.

NeuroML can also model the individual electrical coupling, synaptic transmission together with detailed models of individual neurons
The validity of NeuroML-based neural network model descriptions were established when simulations showed that the models exhibited a similar behaviour across a number of independently developed simulators. However, results show that the simulations slightly differ from simulator to simulator which indicate the limitations of the interoperability of one simulator to another, but this difference arose only at high levels of temporal and spatial discretisation.

Therefore, the development of NeuroML as a simulator independent description language for neural network models allows better interoperability, again making it easier to evaluate, access and reuse different models.

Below is an extract of code of how the low-level, procedural PyNN interface is used in a simple example where a network comprising of an integrate-and-fire (IF) neuron receiving spiking input from a Poisson process is constructed. Along with the code a commentary of what each section of code executes and it’s role is provided below.

First, we choose which simulator to use by importing the relevant
module from PyNN:

```
>>> from pyNN.neuron import *
```

If we wanted to use NEST, we would just import *pyNN.nest*,
etc. Regardless which simulator back-end we were to choose, the following code will remain the same.

```
>>> setup(timestep=0.1, min_delay=2.0
```

After we have set the global parameters of the simulator using the above line of code.
We then proceed to create the following two neuron types:

- An IF neuron with synapses that respond to a spike with a step increase in synaptic conductance, which then decays exponentially.

- A “spike source”, a simple neuron that emits “spikes” at predetermined times however is incapable cannot receive input spikes.

```
>>> ifcell = create(IF_cond_exp,
... {'i_offset': 0.11,
... 'tau_refrac': 3.0,
... 'v_thresh' : -51.0})
>>> times = map(float, range(5,105,10))
>>> source = create(SpikeSourceArray,
... {'spike_times': times})
```

the *create()* function deciphers the standard Pynn model name, *IF_cond_exp* for this situation, into the model name utilized by the simulator, *Standard_IF* for NEURON, *iaf_cond_exp* for NEST, for instance furthermore interprets parameter names and units into test simulator particular names and units. To take one example, the *i_offset* parameter represents the amplitude of a steady current infused into the cell, and is given in nanoamps.

The similar parameter of the NEST *iaf_cond_exp* model has the name *I_e* and units of picoamps, so Pynn both converts the name and duplicates the numerical esteem by 1000 when running with NEST. Standard cell models and programmed interpretation are examined in more detail in the following area.

The *create()* capacity gives back an ID object, which gives access to the parameters of the cell models, e.g.:

```
>>> ifcell.tau_refrac
3.0
>>> ifcell.tau_m = 12.5
>>> ifcell.get_parameters()
{'tau_refrac': 3.0, 'tau_m': 12.5,
'e_rev_E': 0.0, 'i_offset': 0.11,
'cm': 1.0, 'e_rev_I': -70.0,
'v_init': -65.0, 'v_thresh': -51.0,
'tau_syn_E': 5.0, 'v_rest': -65.0,
'tau_syn_I': 5.0, 'v_reset': -65.0}
```

Having made the cells, we associate them with the *connect()* function:

```
>>> connect(source, ifcell, weight=0.006,
... synapse_type='excitatory', delay=2.0)
```

Now we tell the system what variable or variables to record, run the simulation and finish.

```
>>> record_v(ifcell, 'ifcell.dat')
>>> run(200.0)
>>> end()
```

The consequence of running the above model is indicated in Figure 1, which likewise demonstrates the level of reproducibility reachable between diverse simulators for such a straightforward system. The low-level, procedural interface, utilizing the *create(), connect()*

and *record()* functions, is helpful for basic models then again when porting a current model written in an alternate language that uses the make/join maxim. For bigger, more perplexing systems we have discovered that an object oriented methodology, with a larger amount of deliberation, is more viable, since it both clarifies

the applied structure of the model, by concealing execution points of interest, and permits off camera advancements.

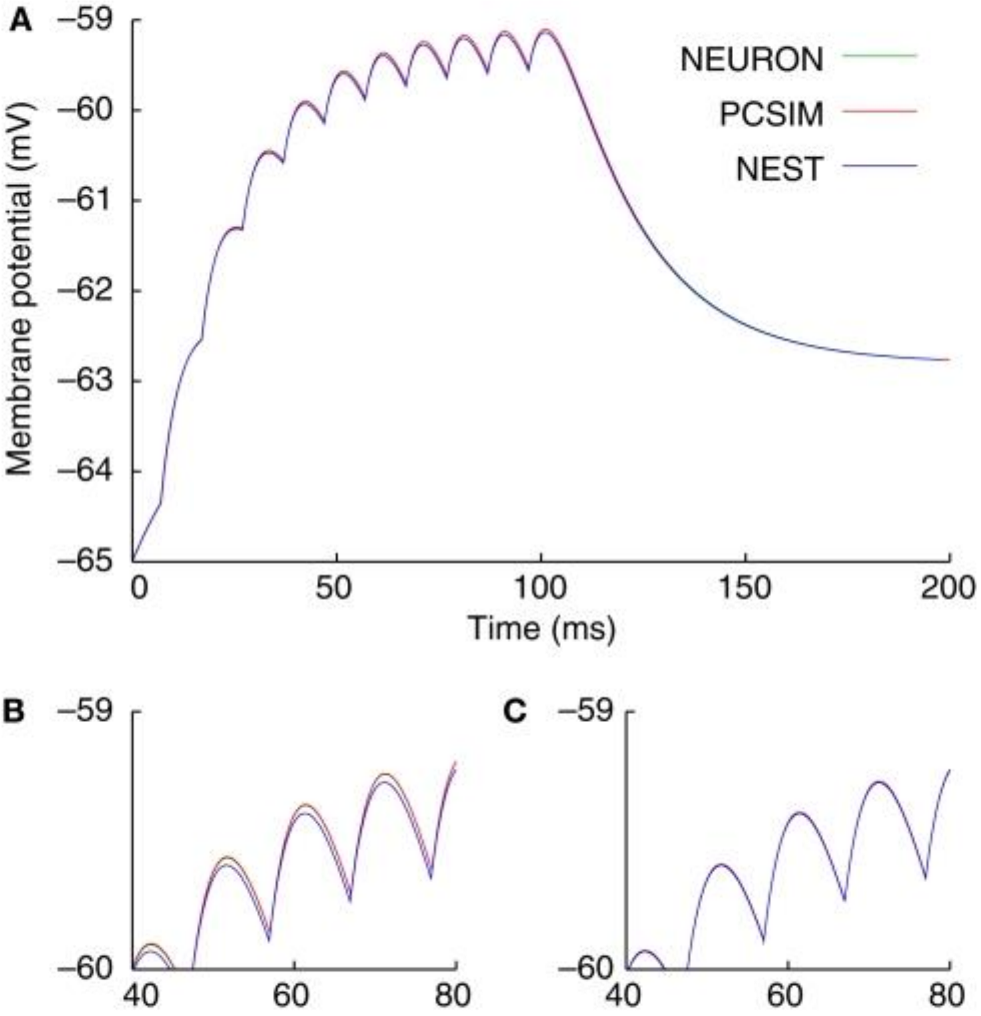

Fig.5 Results of running sample given in the content, with NEURON, NEST and PCSIM as back-end test systems..(A) Entire membrane potential trace with integration time-step 0.1 ms. (B) Zoom into a smaller region of the trace, showing small numerical differences between the results of the different simulators. (C) Results of a simulation with integration time-step 0.01 ms, showing greatly reduced numerical differences.

1.3 Implementation

In this final chapter I will aim to piece the story together and show an example of how one can take these two areas and merge them.

In Chapter 1 we found that the brain is amazing but also big and very complex, then from Chapter 2 we found ways in which to simulate the behaviour of neurons- the very basic component in the brain- in a population. So we find ourselves facing the problem that it is difficult to scale the brain on a meaningful scale. The conclusion is that, if we wish to fully understand how the brain represents and processes information, we have to assemble a supercomputer model to test theories of how the brain functions.

This leads us to the spiking neural network architecture (SpiNNaker) project which is motivated by the grand challenge of understanding how information is represented and processed in the brain [1]. The SpiNNaker machine is a computer designed specifically to support the sorts of communication found in the brain.

The basis of SpiNNaker lies in the idea that we are using massively parallel computation and for this, the architecture has been designed to support a million microprocessor cores. The particular difference between spinnaker and other supercomputers is that it is "optimized to carry very large numbers of very small packets" as opposed to the traditional supercomputers systems which focus on an optimisation of large data packets.

In the particular case that we are focussing on in the cerebellum where we would like to model a system with an arbitrary interconnectivity such as the microzone. SpiNNaker utuilizes PACMAN which is essentially a software layer that allows us to write our models on a simulator of our choice interpret it and run on SpiNNaker. The main role of the software is to maintain it's flexibility for users with particular models because as we mentioned before different simulators can be chosen simply depending on how best they complement your particular model. Therefore making as simple as writing your model and in a popular language such as PyNN and having PACMAN decipher it for you.

A straightforward illustration system is shown in Fig. 8 (left): excitatory and inhibitory populaces are repetitively interconnected. The proportion of excitatory to inhibitory neurons is situated to 4 : 1 to maintain a harmony between excitation and inhibition.

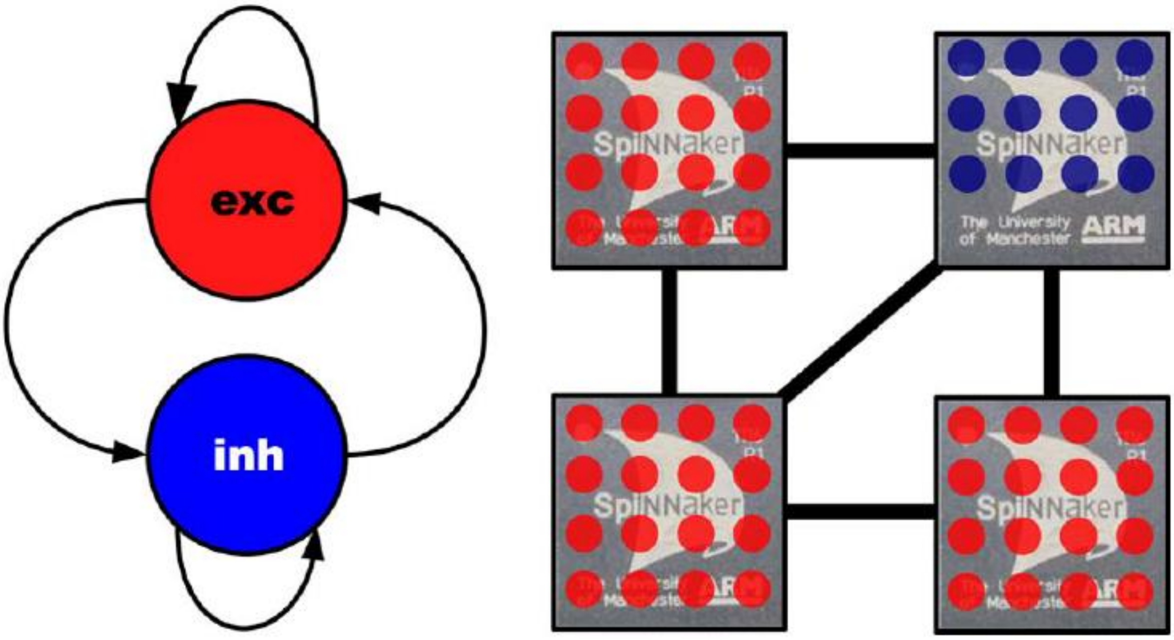

Fig.6 Illustration system (left) with one excitatory and one inhibitory populace with a size degree of 4 : 1 is mapped by PACMAN onto 60 processors on four Spinnaker chips (right).

So we see also here that the need for the super computer we required capable of modelling to our needs is met in the form of SpiNNaker. Large scale of course however with the capability of configuring a million core computer we find we have each core modeling upto a thousand neurons and a million synapses, which poses an obstinate dilemma as one billion neurons need to be mapped and one trillion synapses need to be routed to implement a user-specified model. However, as discussed above the introduction of PACMAN not allows us to use the familiar PyNN but also helps us with this mapping problem.

As we see currently SpiNNaker is showing all the right signs to become even more competent however of course when attempting to improve such a system they are likely to face are in observability of movement inside the machine, application mapping and loading performance, however most strikingly with debugging vast scale models running on the machine.

On a further note the Human Brain Project (HBP) is worth mentioning as it is an example mass scale project to essentially uncover how the brain works by employing specialists from different disciplines such as Biolgy and ICT to merge their efforts and turn the goal of understanding the human brain into a reality!
We here at the Technishe universität München chair of Neuroscientific theory are also working on a sub-project part of the HBP in motor movement of robotic arms by modelling the Cerebellum. SpiNNaker is so widely used amongst the computing projects.

Conclusion

In conclusion, I believe that there is certainly a justifiable need to commence research into the human brain as a result of its sheer importance in our everyday lives. We come across many cases of people with damage to the cerebellum and interestingly the same incident a severe "knock to the head" e.g. can impair a patients in different ways depending on the part of the cerebellum damaged. We also now know a lot a lot the biology that is present and can reduce the minimum functional unit of the cerebellum down to a microzone. Fortunately, we found ways to model neural networks as accurately as possible but there also limitations to our models because most neural network simulators are based on neuron populations in which the neurons within them spike but numerous biological subtle elements are precluded in this expansive framework, e.g. that the firing of a neuron may be deterministic or stochastic depending on its internal state as evidence from neuroscience clearly suggests. However, this is not necessarily fundamental to computing. Nevertheless, it's vital that we know that although we are making big steps and fantastic progress with innovative projects such as the aforementioned SpiNNaker etc. but we still have to remember they are not perfect and lack some subtle details. Not least to mention that even the HBP project has deployed an ethics team, of which some pose the argument of whether or not it is morally correct to attempt to model a human brain at all.

On a more positive note though I feel it is a great opportunity to intertwine disciplines and engage in critical thinking with people from all different backgrounds and regardless of whether or not the brain is eventually recreated or modelled we will have certainly gained a better insight into how we can improve quality of life whether it be in the field of medical technology to help patients with damage to the brain.

Student Reflection

The student reflection chapter I decided to include in order to talk specifically about my personal experience during these 11 weeks, what tasks I participated in, my individual assessment on my progress and how I feel it has impacted and prepared me for future opportunities.

In the first few weeks I was introduced to my supervisor and the different research projects he was conducting and I was assigned to help assist him on the aforementioned HBP project in which we are using the cerebellum as a biological model, so my first assignment was to research various aspects of our model of the cerebellum and report back with suggestions on how we could make the model more biologically accurate. And we succeeded in mimicking in our model inhibitory neurons to map input from “mossy fibers” to the “granule cells” via iteration tools method which helped to make our more model more representative of an actual system.

As a result about halfway through I was then instructed to give a presenation to seniors and fellow students at the Chair on the Cerebellum and why we took a particular interest in it and our progress so far. Which again was successfully received. In this time after about 6/7 weeks in as well as continuing on with research, I was then required to attend various important talks and meetings around the city on important topics closely related to our work from popular German speakers in the field, sometimes with and without my supervisor. Meanwhile I was also working on programming learning in Python and trying to run them on PyNN. Towards the end of my time I was working first hand on the robotic arm and running different programs on the bot with my supervisor and getting it working.

All in all I believe that it has been an incredible experience, and has been an eye opening experience. I have seen and worked firsthand on projects, learnt how to do regular intensive research and reporting back in the form regular meetings with professionals, conversed with high profile professors and picked their brains over innovative ideas and technologies. Which in turn have sparked a massive interest in pursuing and furthering my work in this field leading to me actively now searching for a postgrad post in the area of neuroscience and neuroengineering.

List of References