

---

# **Computational Lab Notebook**

AdEx implementation in NetPyNe

Javier Palma Espinosa

Last Revision: 2022-02-26

## Contents

<b>1</b>	<b>Information</b>	<b>3</b>
1.1	Project name . . . . .	3
1.2	Project date . . . . .	3
1.3	Motivation . . . . .	3
1.4	Publications . . . . .	3
1.5	Digital verification . . . . .	3
1.6	Links . . . . .	3
1.7	Additional Credits . . . . .	4
<b>2</b>	<b>Changelog</b>	<b>5</b>
<b>3</b>	<b>Simulation Workflow</b>	<b>6</b>
<b>4</b>	<b>Daily Report</b>	<b>7</b>
4.1	Jun 7th, 2021 . . . . .	7
4.2	Jun 30th, 2021 . . . . .	7
4.3	Jul 6th, 2021 . . . . .	7
4.4	Jul 19th, 2021 . . . . .	7
<b>5</b>	<b>Preliminary Results</b>	<b>9</b>
5.1	Objectives . . . . .	9
<b>6</b>	<b>Meetings</b>	<b>10</b>
<b>7</b>	<b>Computing tools, code snippets, and tips.</b>	<b>11</b>
7.1	CODE: Fast prototyping in code (2021-12-27) . . . . .	11
7.2	CODE: git store credential . . . . .	11
<b>8</b>	<b>Papers summary</b>	<b>12</b>
8.1	Selective attention model with spiking elements . . . . .	12
	<b>References</b>	<b>14</b>

## 1 Information

### 1.1 Project name

Adaptative Exponential Integrate and Fire (AdEx) implementation in Netpyne

### 1.2 Project date

This project started in June 2021

### 1.3 Motivation

The [Kerr Lab](#) implemented a previous version of the AdEx model. The basic idea here was to implement an AdEx Class.

**Should netpyne provide a basic NEURON models, such as AdEx, izhikevic, integrate and fire?**

**Do NEURON provide those models?**

To solve those questions, I propose to implement the AdEx model in NEURON and then, plug it into NetPyNe, as a new functionality.

### 1.4 Publications

1. (Dura-Bernal et al., 2019)(<https://elifesciences.org/articles/44494>)

### 1.5 Digital verification

not yet

### 1.6 Links

- The NetPyNe project is available at <http://www.netpyne.org/>
- Dura-Bernal Laboratory could be reached at <http://dura-bernal.org/>
- The whole project is archived in github as digital repository. It may be found in the following link: <https://github.com/jpalma-espinosa/netpyne>

## 1.7 Additional Credits

- This document was done with the [Eisvogel Template](#), by Pascal Wagler

## 2 Changelog

Date	Commit	log
2022-02-26 20:29:36	<a href="#">cb63c70</a>	update in project structure. New documentation available
2021-07-20 20:44:24	<a href="#">10a7bcb</a>	Pseudo working example of Adex2021b
2021-07-07 05:46:44	<a href="#">4601b03</a>	Changed typo in README
2021-07-07 05:35:18	<a href="#">be151ad</a>	Upload new documents. Thoughts and ideas in README
2021-07-01 06:40:27	<a href="#">026e684</a>	Notes name changed
2021-07-01 06:30:44	<a href="#">ffda310</a>	Ball-Stick class is created (not working).

### **3 Simulation Workflow**

## 4 Daily Report

### 4.1 Jun 7th, 2021

I was able to run the [izhikevic tutorial](#). Also, I wrote the Adex.mod file, by replicating what was done with [izhi2007b.mod](#). However, I am still not able to produce a spike in the Adex model. The izhikevic one has some strange way of calculating the derivative states. What is the difference between those two forms of calculation? Also, how can I incorporate the synapses in the Adex neuron?

The izhikevic (and adex) is implemented as a POINT PROCESS (see also [NEURON documentation](#)), contrary to the HH model.

### 4.2 Jun 30th, 2021

I was on halt because I had to deal with my master thesis. I am now a Master of Science :D.

Because the previous implementation wasn't successful, I asked whether AdEx should be defined as a mechanism or a point neuron (see De Schutter book, Ch. 7). The way that NEURON is implemented, makes logical to define AdEx as a point process and define it as ARTIFICIAL\_CELL. To do this, I have to understand how NET\_RECEIVE (w) process works.

This code block is better defined in the Neuron Book (Ch. 10)

### 4.3 Jul 6th, 2021

I only read a couple of documents from [Neuron tutorial](#) and from a [MIT tutorial on Neuron](#). The important part here was to examine how to properly define the punctual neuron AdEx. It seems that my model needs to considerate an external current *FROM* an external point mechanism. For this, I will need to re-study the [integrate and fire model](#) that is proposed in the [Neuron Github page](#)

### 4.4 Jul 19th, 2021

After the meeting with Salvador, on Jul 6th, and by following his advices, I replicated what was developed in the izhikevich model. In particular the b part. Briefly, the models could be summarized as:

Characteristic	Izhi2003a	Izhi2003b	Izhi2007a	Izhi2007b
Kind	P.Proc.	P.Proc.	P.Proc.	P.Proc.
Section	Dummy	Regular	Dummy	Regular

Characteristic	Izhi2003a	Izhi2003b	Izhi2007a	Izhi2007b
Synaptic input	yes	no	yes	yes
Synaptic method	$g'_{syn} = -g_{syn}/\tau_g$	—	AMPA/NMDA/GABAell dynamics	dependent
Implemented in Netpyne	no	no	yes	no

I focused on replicating the Izhi2007b.

### Results:

1. I was able to build and compile Adex2021b (I am keeping the name scheme). 2. I was able to replicate the Izhikevich tutorial, but now using Adex (adex.ipynb).

### Drawbacks:

1. my neuron does not fire, even further, I get an error

See <http://neuron.yale.edu/neuron/credits>

```
loading membrane mechanisms from
↳ /home/javier/Neuroscience/netpyne/AdEx/x86_64/.libs/libnrnmech.so
Additional mechanisms from files
"./mod/adex.mod" "./mod/izhi2007b.mod"
nrniv: unable to open font "*helvetica-medium-r-normal*--14*", using "fixed"
oc> -65
Segmentation violation
Backtrace:
terminate called after throwing an instance of 'std::regex_error'
what(): regex_error
Aborted (core dumped)
```

I need to debug the .mod file, but I don't know how



## **5 Preliminary Results**

### **5.1 Objectives**

## 6 Meetings

## 7 Computing tools, code snippets, and tips.

### 7.1 CODE: Fast prototyping in code (2021-12-27)

One of my biggest mistakes is to try to build, at first, a very complicated piece of software, which is amendable for programmer, but also efficient as hell. **I must focus on building a working piece of software and then improve it!**

### 7.2 CODE: git store credential

general formula:

```
// local
git config credential.helper store
// global
git config --global credential.helper store
```

```
$ git config credential.helper store
$ git push http://example.com/repo.git
Username: <type your username>
Password: <type your password>
```

Notice that **Password** is the code obtained from the [github access token](#) *several days later*

```
$ git push http://example.com/repo.git
[your credentials are used automatically]
```

## 8 Papers summary

### 8.1 Selective attention model with spiking elements

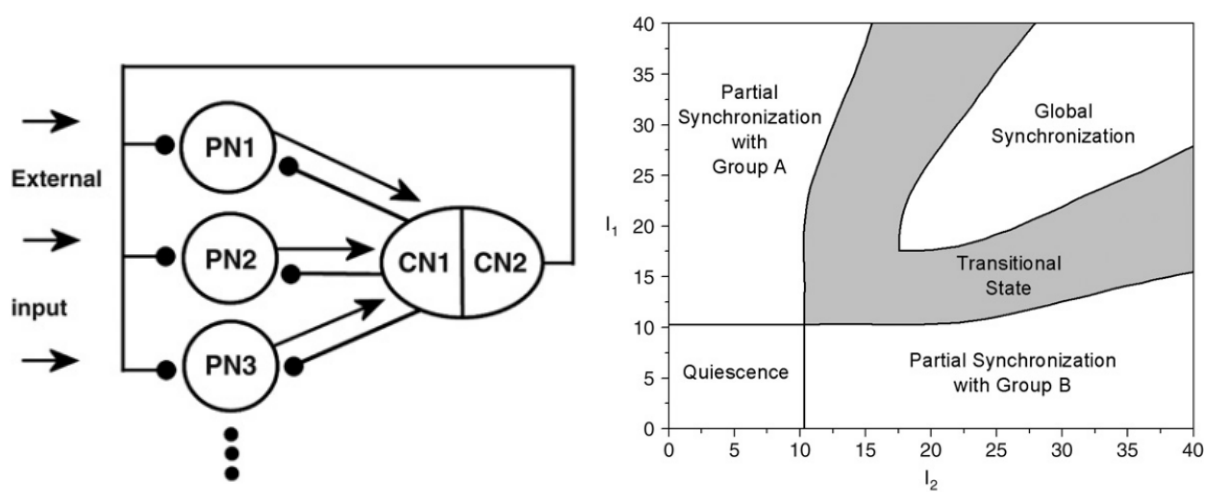
#### 8.1.1 Introduction

1. Visual attention in human and monkey brains is realized by a large-scale distributed neural network that includes several cortical and subcortical areas with bottom-up and top-down flow of information between them.
2. **Despite intensive studies of neuronal activity related to attention, it is still unclear what neuronal mechanisms are used by the brain to implement attention.**
3. Two types of attentional modulation has been proposed:
  - Increased excitation of neurons representing attended stimuli is observed while neural activity evoked by unattended stimuli is reduced to a low level.
  - Gamma range oscillations correlate with the activity of neurons in the attentional focus.

*The model proposed by the authors, using Hodgkin-Huxley neurons, aims to elucidate how selective attention can be represented by the synchrony and suppression of neural activity in a network of interactive spiking elements*

#### 8.1.2 Results

1. The model exhibits five global dynamical states: Partial synchronization A; Transition state; Global synchronization; Partial synchronization B; Quiescence.
  1. Partial synchronization is interpreted as selective attention, where population A or B is “attending” to some stimuli.
  2. Transition state is interpreted as different degrees of attention concentration (could be attentional shift?)
2. **There remain many uncertainties about the complete bifurcation structure of the model, therefore further investigations are required**
3. In the simulations, selective attention (associated with partial synchronization) always favours a group with higher frequency



**Figure 1:** Model(left) and bifurcation space(right)

## References

Dura-Bernal, S., Suter, B. A., Gleeson, P., Cantarelli, M., Quintana, A., Rodriguez, F., et al. (2019). Net-PyNE, a tool for data-driven multiscale modeling of brain circuits. *Elife* 8, e44494.