

Spatial Pattern Optimization for Intra-Cortical Stimulation with High Density MEA

Background

The capabilities of invasive *in-vivo* neural recording-stimulating systems made of multiple contact, so called Multi Electrode Arrays (MEA), is improving lately, mainly because of advances in fabrication processes and miniaturization capabilities. While there are many examples of successful very high density MEA developed for *in-vitro* recordings (Berdondini, other ref, ecc), the challenge of *in-vivo* recordings lies in the fact that there are much stricter size and power consumption limitations. New prototypes have been developed in the last years which attempt to overcome this limitation and to reach *in-vitro* high density also for *in-vivo* systems. This challenge is well motivated: such micro system, in fact, have the potential of extracting information not only about some neurons like tetrodes, but they could record, identify, and even localize hundreds of neurons in the proximity of the electrodes (localizing ref). The problem of going in the tens of micro-scale spacing, when dealing with neural recording, is mainly about noise. The smaller the electrodes are, the higher the noise; with such a diminished SNR it gets more challenging to identify spike waveforms and the spike detection-sorting can result in accuracies. On the other hand, though, the higer density of electrodes assures a high number of simultaneous recordings, giving the possibility to reduce noise by applying array processing techniques (e.g. spatial averaging). The integration of these high density MEA in CMOS technology can be achieved through the use of Electrolyte-Oxide-Semiconductor Field-Effect-Transistor (EOSFET). The gate of the transistor itself serves as sensor and it is coupled through a dielectric layer in order to increase the capacitive component. Purely capacitive contacts, could also be used to stimulate the tissue with different approaches. One of the latest example is given by the ENIAC chip (ref), a closed-loop fully encapsulated chip for neural recording and stimulation, which applies tri-phasic constant currents pulses by providing balanced voltage ramps. Another recent prototype worth mentioning was developed by (thewes ref): a 16 by 16 matrix of EOSFET with 15 μm pitch was fabricated on a single shank of 300 μm width. The system only performed recording (no stimulation) and it was tailored for Local Field Potentials (LFPs), but integration with stimulation could be possible either with a switching system and direct access to the gate of the EOSFETs, or with a parallel circuitry and interleaved electrodes. The former approach would not allow simultaneous recording and stimulation, but it would yield a higher resolution, whereas the latter would be capable of simulataneous recording and stimulation at the expense of contact density.

A system capable of both recording and stimulating neural tissue is very interesting especially in order to perform closed-loop experiments, in which stimulation is triggered from certain features extracted from recording the same neural sites. Such implants become very fascinating because they have the capability of inducing neuroplasticity *in-vivo*, i.e. they can make neurons learn. Jackson et al. showed that by stimulating for 48 hours a *target* neuron in the motor cortex 5 ms after a second *trigger* neuron fired an action potential, the *trigger-to-target* connection was greatly enhanced up to 10 days after conditioning was stopped. Many following studies tried to use these spike-triggered stimulation to improve neural rehabilitation and recovery (cortico spinal fetz, nudo).

The main problem of recording-stimulation systems is the asymmetry between the two modalities. While for

recordings it is relatively easy to reach a single-neuron discrimination by using spike sorting techniques, that allow to identify different neuron based on their recorded waveform features, stimulation is much more rough. Hundreds of μA are injected (or drawn) from electrodes and they spread radially in all directions. This results in a diffuse activation of neurons surrounding the stimulation site. Rebesco et al. demonstrated that by applying spike-triggered stimulation in a rat somatosensory cortex, not only the connection between *trigger* and *target* was strengthened, but also many *non-target* connections were affected and the entire neural network of recorded neurons changed its behavior.

When studying neuroplasticity, it is definitely of great importance to be as selective as possible in order to be able to limit parasitic effect and to have a better controlled environment. Recent studies have demonstrated great promise, showing that from MEA recordings it is possible to localize neuronal somas in the surrounding of the electrodes. In Ruz et al., 144 neurons were localized (soma position) *in-vivo* with a 32 electrode MEA with an area of $275 \times 36 \mu\text{m}^2$ (spacing $22\text{-}25 \mu\text{m}$). From another recent work (ref) performed with high density (HD) MEA ($17.5 \mu\text{m}$ pitch) *in-vitro*, not only neurons were localized, but such a high density was also used to reconstruct the axonal arbors using the redundancy of recordings and spike-triggered average.

Certainly, *in-vivo* recordings are more noisy and less controllable, but it may be possible to exploit HD MEA to estimate the position of neurons surrounding the MEA and also to estimate the direction of propagation of the axon hillock, i.e. the axon tract more proximal to the neuronal soma. In this work we assume that the previous information is available and we exploit the same MEA to tailor stimulation spatial patterns to be as selective and as effective as possible. The goal is to show that spatial patterns can be optimized in order to depolarize/hyperpolarize specific identified neurons, while not activating others.

We show that it is possible to selectively activate neurons which are as close as $5 \mu\text{m}$ from each other (much less than actual cortical separation), but with a different axon direction. The main limitation of this approach, though, is that in order to keep optimization fast and implementable online, simple models are used for the MEA stimulation, which introduce estimation errors, that should be quantified with more sophisticated models.

Methods

Models

The computational framework for performing the simulations was completely developed in Python, using custom models for the MEA and LFPy (ref), based on Neuron (ref), for the simulation of neural activation and responses.

The **MEA** was modeled as a grid (10×10) of monopolar current sources on a semi-infinite plane, with a pitch of $15 \mu\text{m}$, resembling the prototype described in (thewes). The semi-infinite plane approximation os due to the fact that electrodes lie on a shank, facing the neural tissue only from one side. The maximum current deliverable from each electrode was set to $50 \mu\text{A}$. With this model, the potential at position \vec{r} is:

$$V(\vec{r}) = \sum_i \frac{I_i}{2\pi\sigma |\vec{r} - \vec{r}_i|} = \sum_i \alpha_i I_i$$

where \vec{r}_i and I_i are the position and current of the i -th electrode and σ is the conductivity of the tissue, assumed to be isotropic and homogeneous. This model is clearly to simple to represent the reality of stimulating a tissue especially because the tissue is not homogeneous and isotropic as there are a multitude of dendrites, axons, glial cells, vessels, etc. On the other hand, this model is very simple and linear in the currents, which makes it very interesting and fast for optimization problems.

The **neurons** were modeled in a different way for optimization and evaluation. Neuron cables are usually modeled with the so called cable equation, which describes the spatio-temporal dynamics of a linear tract of

cylindric made of discrete compartments. When the length of each compartment tend to 0, the cable equation can be expressed in a continuous form:

$$\lambda^2 \frac{\partial^2 V_m}{\partial x^2} + \tau_m \frac{\partial V_m}{\partial x} - V_m = -\lambda^2 \frac{\partial^2 V_e}{\partial x^2}$$

in which λ and τ_m are the spatial and temporal constant, respectively; V_m and V_e are the membrane potential, and the external potential and x is the axon direction. The right-hand side of the cable equation governs the generation of spikes and can be used to predict their location along the axon. For this reason it is referred as *activation function* (AF) (Rattay 1990).

In order to excite the neurons, thus, the second derivative of the potential along the axon direction must be higher than a certain threshold. For optimization of the spatial pattern a neuron is represented by a single segment starting from the soma location, with the direction and length of the axon hillock. Hence a *geometric neuron* consists of a 3D point (soma), an alignment (axon hillock direction), and a length (which was set to 15 μm). Moreover, since the second order derivative must be computed numerically along the 3D axon direction, also a number of discrete points along the axon is specified and it was set to 3 (which basically gives a positive 2nd derivative if the central potential (at 7.5 μm) is smaller than the edges (0 and 15 μm), and viceversa).

When the optimal spatial pattern is generated, it is evaluated on more sophisticated neuronal models in the LFPy environment. From linear passive models implementing the cable equation, to multi-compartmental models with more complex geometry, passing through simple linear active models (WORK IN PROGRESS). In this way we can evaluate the efficacy of the optimization and the simple model described above is sufficient to target and activate neurons lying in the surrounding of the MEA.

Optimization

The optimization framework is implemented using DEAP package (ref). Genetic algorithm have been selected for different reasons:

- the high number of degrees of freedom (in the 10x10 configuration there are 100 parameters) would make other numerical optimization very time consuming
- it can be easily customized in terms of fitness function, random generation, mutation operations, and constraints
- it is relatively fast and therefore it could be suitable for an online setup

Genetic algorithms perform optimization with a stochastic approach, by randomly sampling the solution space (in this case the 100 currents to be assigned to each electrode) and evaluating them with a fitness function. Therefore, the fitness function plays a very important role and represents the key to achieve a good optimization. Let us summarize what we want to achieve with stimulation so that it would be easier to understand how the fitness is built. First of all, we want to depolarize the target neuron(s) by applying a positive AF along its axon. It should be emphasized that the goal is not to maximize the AF as it would result in a diffuse and not focalized stimulation, but it is enough to exceed the threshold by a safety margin. Second, when other surrounding neurons are present, we want to find a spatial pattern that does not generate a spike. Also in this case we would like a certain margin, so we can set the AF along surrounding axons to be lightly hyperpolarized. Third, we would prefer a sparse solution, for it would be easier to implement and it would consume less power (power efficiency is very important especially for implanted devices). This said, the fitness function have 3 objectives:

1. Activation of target neuron: for all points in which the AF is evaluated, add 1 if it's above threshold, or subtract the difference between the AF value and the threshold normalized over the difference between the activation threshold and the non activation threshold.
2. Light hyperpolarization of surrounding neurons: same as point 1. but inverted (add 1 when AF is below non activation threshold, otherwise add the normalized difference)
3. Solution sparsity: sum of zero-current electrodes

The 3 objectives are maximized and they are given the same weight. The solution will then represent a trade-off among these three objectives .

In order to keep an eye on a physical implementation of the system, some constraints on the value of currents are added. In particular the currents can go from $-50\mu A$ to $+50\mu A$, but the can only have $5\mu A$ steps; therefore currents can have 21 different values, but a negative current would be the opposite of the positive one, so only 10 values (excluding 0) can be generated.

Selection of the solutions to be mated was performed with a selection tournament approach. Two-point crossover was used as crossover operator and crossover was applied with 80% probability. Mutation was applied to 10% of the population and it consisted of 2 steps: in the first step a gaussian mutation was applied with 20% probability to each current value; then, in order to favor sparse solution 10% of the current values were randomly set to 0. The best solutions of each iteration were always kept in the offspring.

The genetic algorithm was run for 50 generations and the population was made of 100 individuals.

Results (preliminary)

Single Neuron

As anticipated in the previous section, the optimization is tailored to deal with multiple neurons (a target and some surroundings). We tried to run the optimization also for a single neuron lying $10 \mu m$ away from the MEA ($x = 10$) and with different orientations in the plane parallel to the MEA ($y-z$ plane). From the cable equation and the AF, the best way to stimulate an axon would be with a cathodic current (durand, rattay), i.e. a sink. Fig 1 shows the potential on the yz plane at $x=10\mu m$ and it can be noticed that the spatial patterns is basically cathodic (blue represents a potential trough). Some spurious currents appear in the left and central panel, but they would have most probably disappear with a higher number of generations due to the sparsity mutation. It should be noticed that at this stage the algorithm does not try to focalize the activation as no information of surrounding neurons is provided. (horizontal inclination is not presented but for symmetry along y and z axis it should give the exact same results).

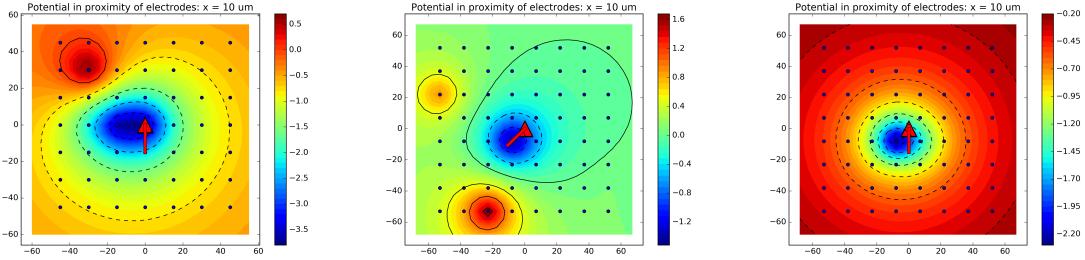


Figure 1: Single neuron activation with different alignments and positions

Multiple Neurons

The application optimization starts to be of interest when we throw in some surrounding neurons. Let us start by adding a neighbor to the target neuron and check the results!

Two Neurons

In Fig 2, 3 different configurations are shown: on the left 2 neurons separated by 20 μm aligned in the z direction; in the center panel the 2 neurons are 5 μm apart and with 45 degrees divergence (they overlap); in the right panel the divergence is only 30 degrees. In all 3 cases the spatial pattern manages to depolarize the target neuron and hyperpolarize the surround neuron, as shown in Fig 3 and 4, respectively. Fig shows a matrix with the current values.

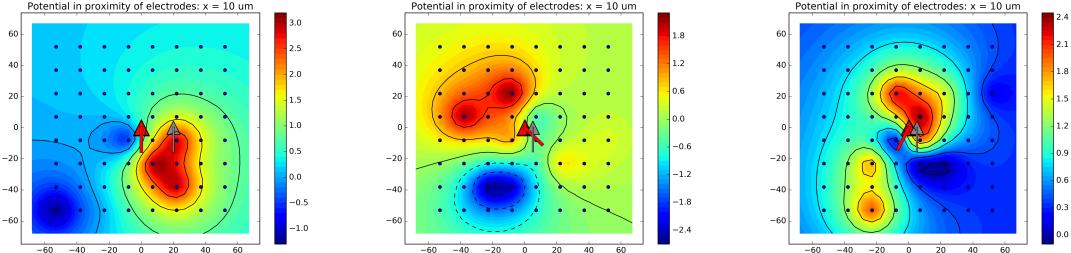


Figure 2: 2 neuron activation with different alignments and divergence

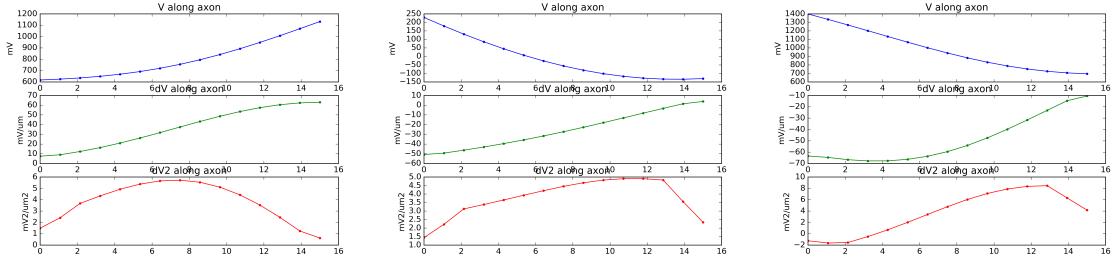


Figure 3: Target activation for configurations in Fig 2

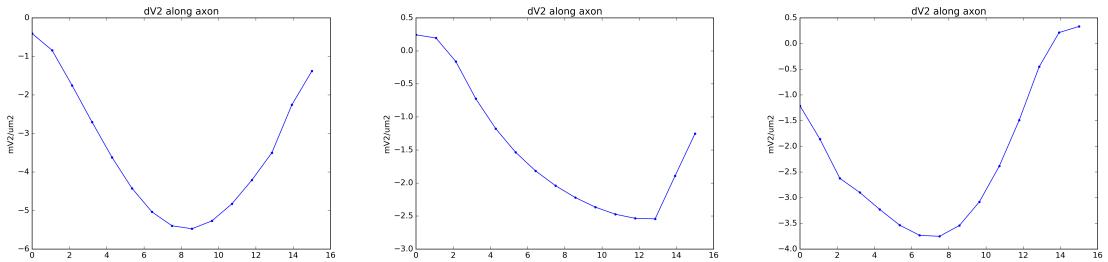


Figure 4: Surround activation for configurations in Fig 2

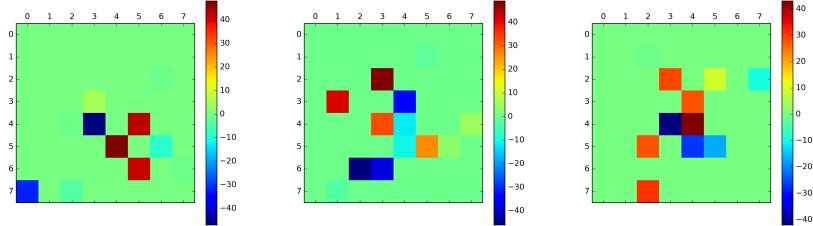


Figure 5: Currents (nA) for configurations in Fig 2

Three Neurons

3 different configurations have been tested involving 3 neurons, and the potential at $x = 10\mu\text{m}$ are shown in Fig. 6. In all 3 cases the algorithm is capable of finding a spatial pattern that meets the requirements, even for highly tangled cases (such as the right panel, in which the target and surround neurons are $5\mu\text{m}$ spaced and with only 30 degrees divergence). In all 3 configurations the targets are reliably depolarized (Fig. 7) and surround neurons fluctuate around zero (Fig. 8), while the solution maintains an significant amount of zero-currents (Fig. 9).

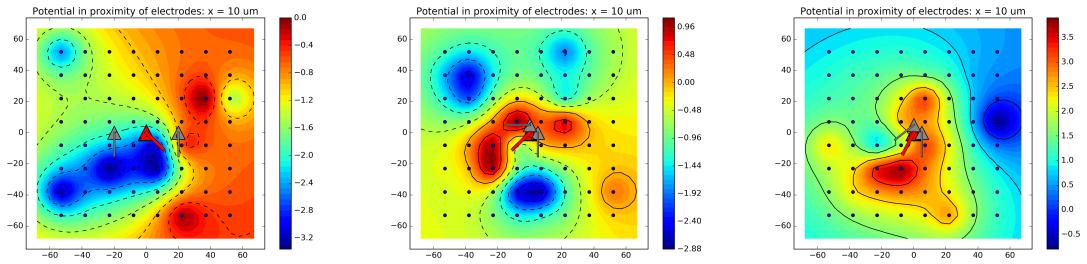


Figure 6: 3 neuron activation with different alignments and divergence

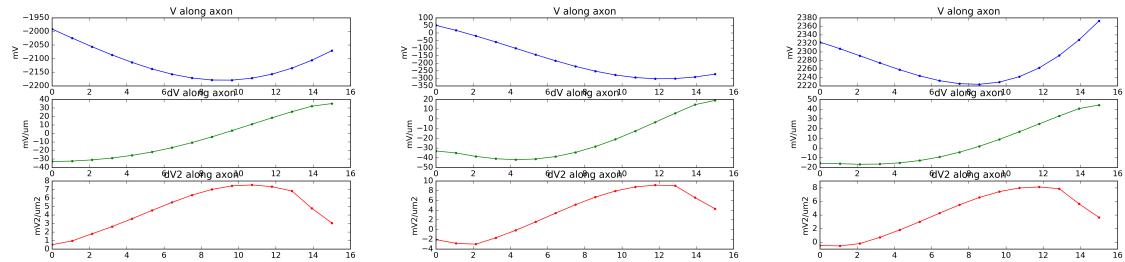


Figure 7: Target activation for configurations in Fig 6

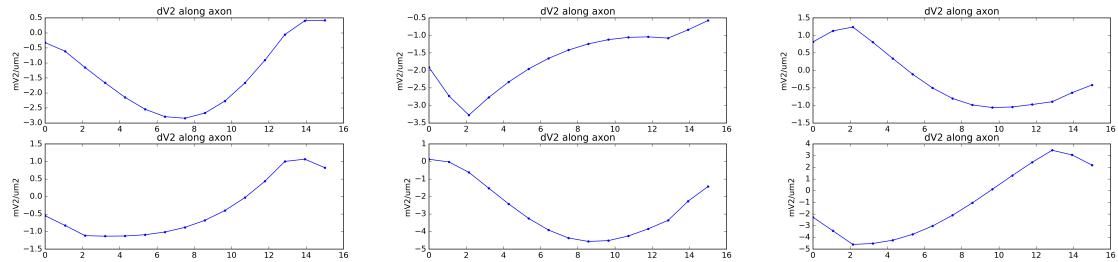


Figure 8: Surround activation for configurations in Fig 6

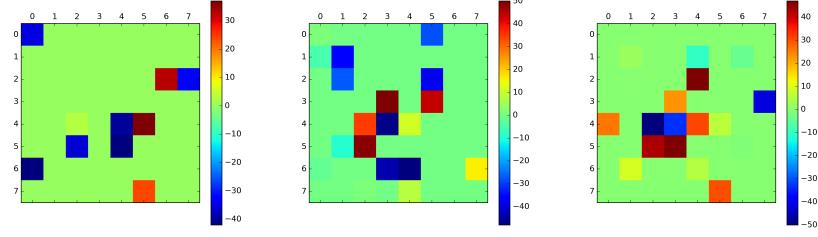


Figure 9: Currents (nA) for configurations in Fig 6

It is also interesting to look at the distribution of the second derivative, i.e. the AF, along different directions. Fig 10 shows how the AF changes along the z, the y axis, and the target neuron axis (built as a weighted average between y and z components based on target alignment) of the second configuration. It can be appreciated visually how the potential is shaped so as to meet the optimization objectives. On the left panel it is clear that the AF is very high for the target neuron.

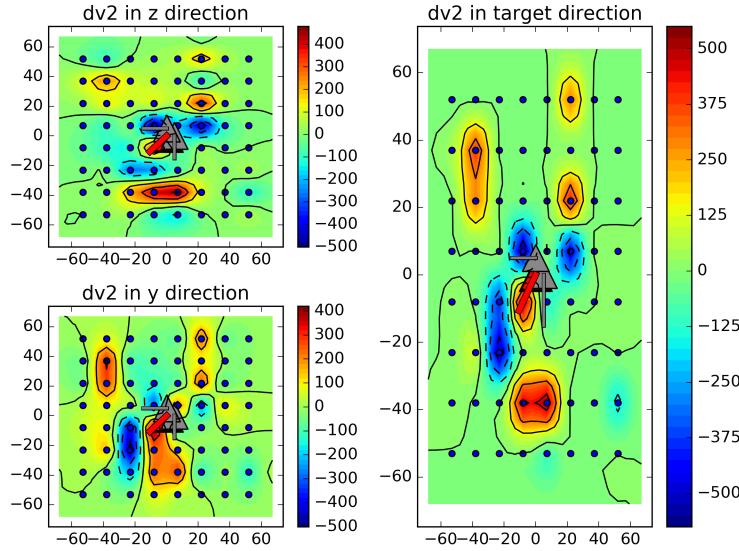


Figure 10: AF in different directions for fig. 6

Five Neurons

An optimization with five neurons with $5\mu\text{m}$ spacing and different orientation have been performed, and also in this case the optimization is well performing. From the potential in Fig. 11 the target neuron can hardly be seen, but from Fig. 12 left panel it can be seen that the target neuron is depolarized, while surround neurons are left in an almost neutral environment (right panel).

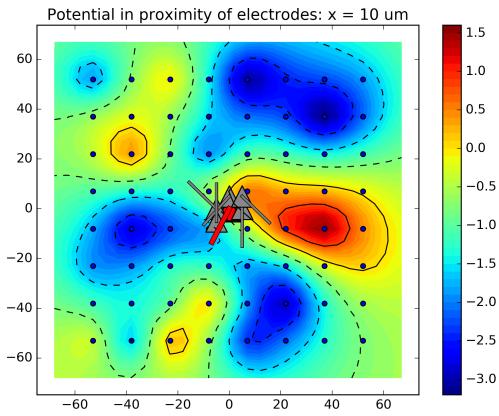


Figure 11: 5 neurons configuration

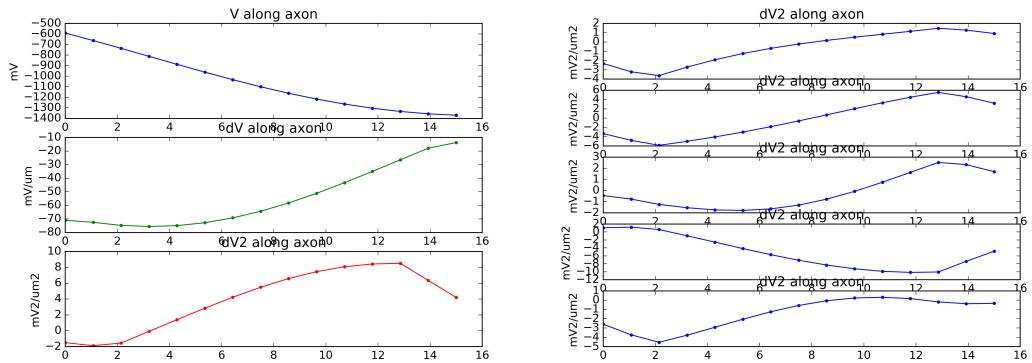


Figure 12: Currents (nA) for configurations in Fig 11

The solution is still sparse enough, as shown in Fig. and the distribution of AF in the different directions is shown in Fig. , but it is very challenging to interpret it due to the intricate scenario.

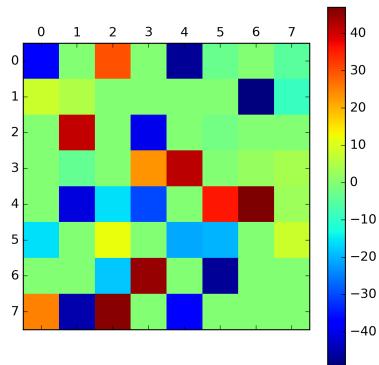


Figure 13: Currents (nA) for configurations in Fig 11

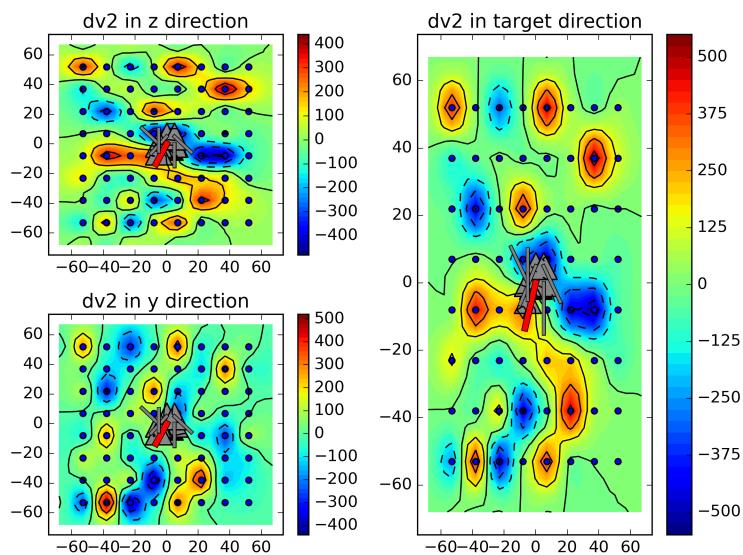


Figure 14: AF in different directions for fig. 11

Discussions

Skype?