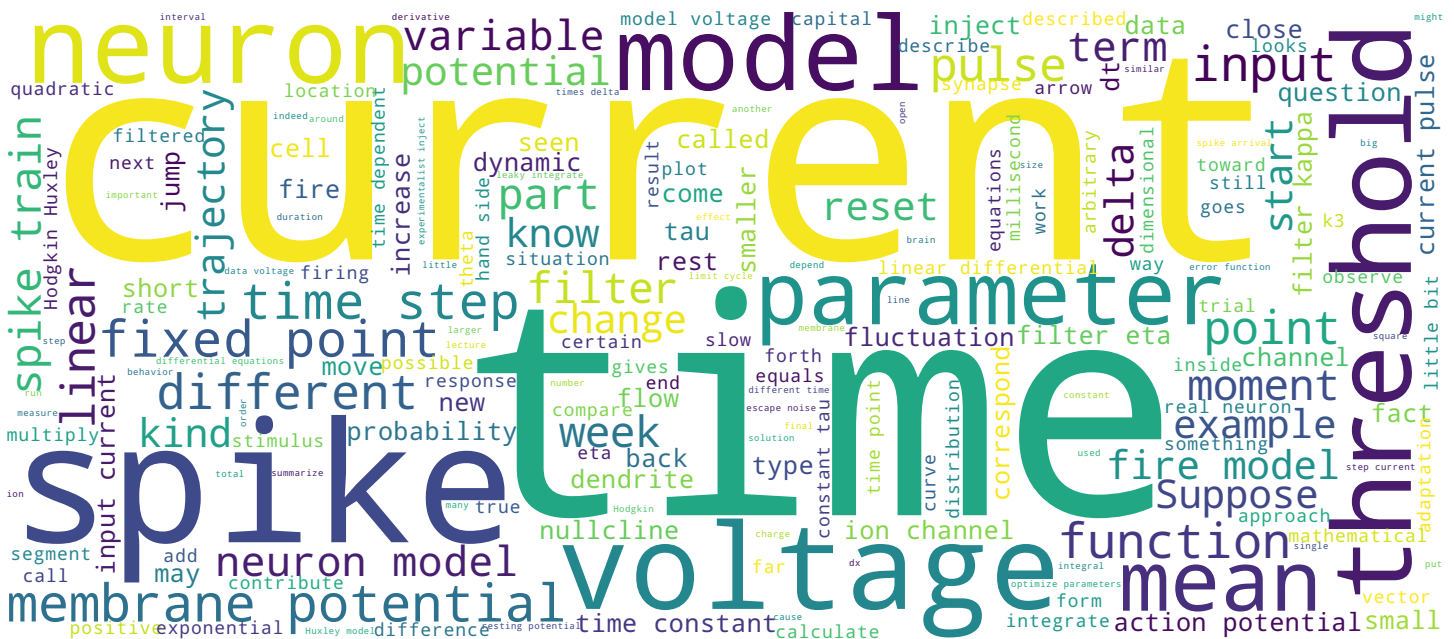


ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

EPFL, Lausanne, Switzerland

- ## 7.7. Helping Humans



EPFL



Week 7 – part 5 : Parameter estimation



✓ 7.1 What is a good neuron model?

- Models and data

✓ 7.2 AdEx model

- Firing patterns and analysis

✓ 7.3 Spike Response Model (SRM)

- Integral formulation

✓ 7.4 Generalized Linear Model (GLM)

- Adding noise to the SRM

7.5 Parameter Estimation

- Quadratic and convex optimization

7.6. Modeling in vitro data

- how long lasts the effect of a spike?

7.7. Helping Humans

Welcome to part 5 of week 7 of the class Neural Dynamics. In the previous part, I reascended the spike response model with escape noise as a realization of a generalized linear model. Let's now talk about how to optimize parameters of such a neuron model.

Notes

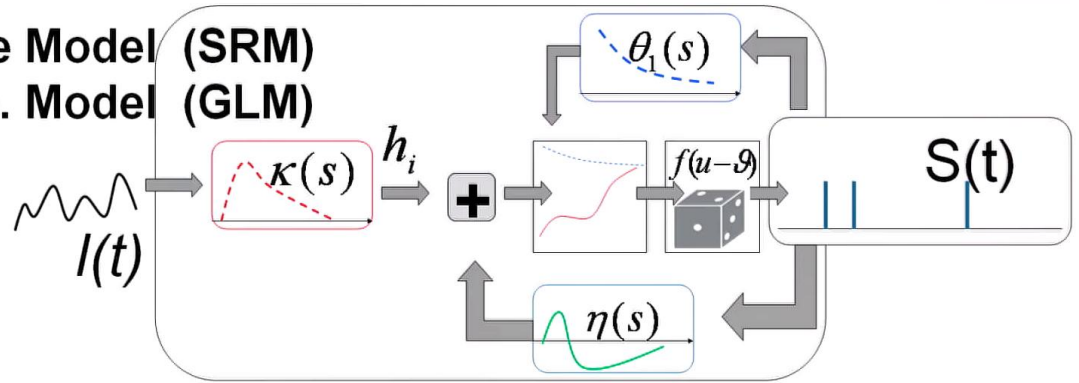
Summary



0m 04s

Neuronal Dynamics – 7.5 Parameter estimation: voltage

Spike Response Model (SRM)
Generalized Lin. Model (GLM)



Subthreshold
potential

$$u(t) = \int \underbrace{\eta(s)}_{\text{known spike train}} S(t-s) ds + \int_0^\infty \underbrace{\kappa(s)}_{\text{known input}} I(t-s) ds + u_{rest}$$

✎ ✎ ✎ ✎

Here again are our model. Input is filtered by a first filter kappa. This is this term over here. Inject a known input. It's filtered with this filter kappa and that contributes to the membrane potential. Then this membrane potential is compared with a threshold, and depending on the distance between membrane potential and threshold, spikes are generated stochastically, and these spikes are fed back by a filter eta, which contributes to the membrane potential. So this feedback is shown here. The spike train is filtered with this filter eta, and that contributes to the membrane potential. So eta is something like the spike after potential. And experimentalist injects the current. So we have a known input, and at the same time, he observes the spike train. The spike train is known as well.

Notes

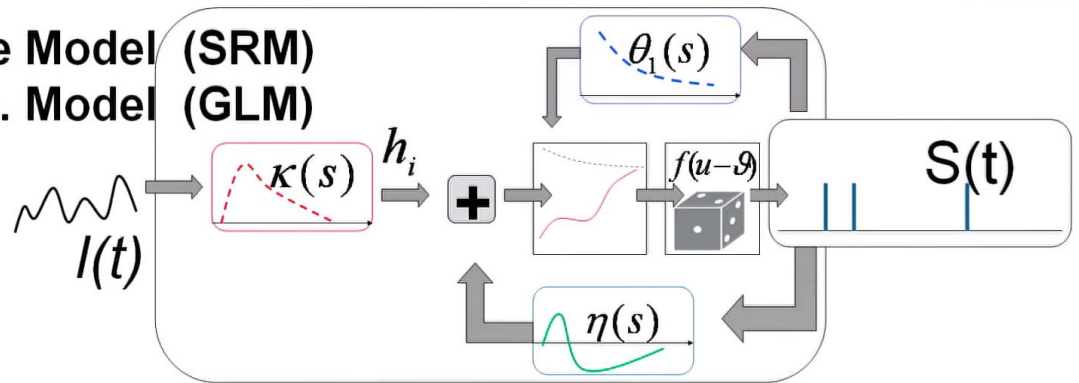
Summary



0m 25s

Neuronal Dynamics – 7.5 Parameter estimation: voltage

Spike Response Model (SRM)
Generalized Lin. Model (GLM)



Subthreshold
potential

$$u(t) = \int \underbrace{\eta(s)}_{\text{known spike train}} S(t-s) ds + \int_0^\infty \underbrace{\kappa(s)}_{\text{known input}} I(t-s) ds + u_{rest}$$

Linear filters/linear in parameters

⌂ / ☐ ➡

Now, the voltage, given this spike train and given input, is just a linear convolution. It's linear in the parameters.

Notes

Summary



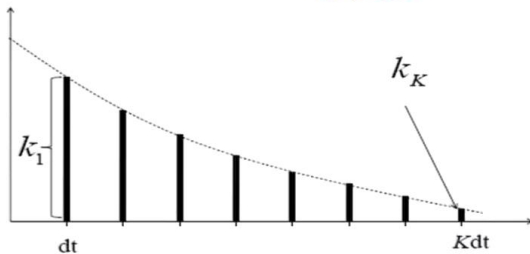
1m 34s

Neuronal Dynamics – 7.5 Parameter estimation: voltage

Linear in parameters = linear fit

$$u(t) = \int_0^\infty \kappa(s) I(t-s) ds + u_{rest}$$

$$u(t_n) = \sum_{k=1}^K k_k I_{n-k} + u_{rest}$$



$$k_k \sim \kappa(1 \cdot dt)$$

$$u^{data}(t)$$

$$u(t)$$

⌂ / ☰ ➡

Let me explain what this means. In practice, we work in this great time. So this integral over past input with a filter kappa is transformed into a sum over time steps. So k is an index, the lower index here, and k runs from 1 to capital K. So k sub 1 would be this first entry. 1 time step later, I have k sub 2. The last entry is k sub k. This is like having a continuous filter shape, kappa of s, and the readout at discrete times. So k sub 1 is proportional to kappa at 1 times dt, where dt is the time step.

Notes

Summary



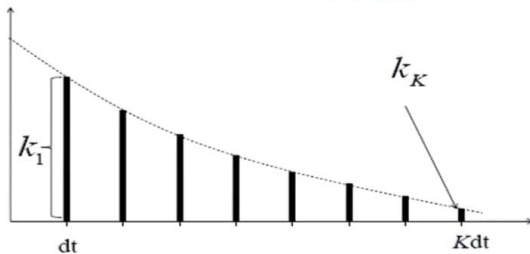
1m 44s

Neuronal Dynamics – 7.5 Parameter estimation: voltage

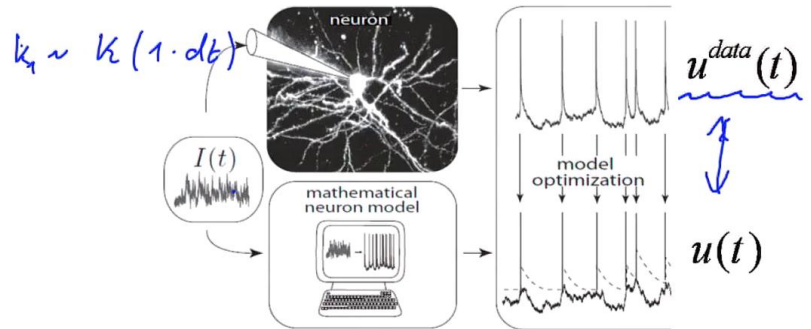
Linear in parameters = linear fit

$$u(t) = \int_0^\infty \kappa(s) I(t-s) ds + u_{rest}$$

model $u(t_n) = \sum_{k=1}^K \kappa_k I_{n-k} + u_{rest}$



comparison model-data



Now, what can we do with this kind of description? Well I said already that the experimentalist injects a current into a real neuron and he measures the voltage. So we have access to the spikes, that's the peaks over here, the action potentials, and we also have access to the voltage trace in between the spikes. So I call this voltage trace u data. Now, as a modeler, I inject the same current into my mathematical neuron model and using this convolution transformed into a discrete time, summation gives me my model u of t . What we need to do is, we need to compare the two. We need to adjust the parameters such that u of t is as close as possible to the measured u . Now, let me come back to the notion of linear in the parameters. I know the current. The voltage of my model is just linear in the parameters. There's no square root of k . There's no k to the cube. It's just linear, and that's important.

Notes

Summary



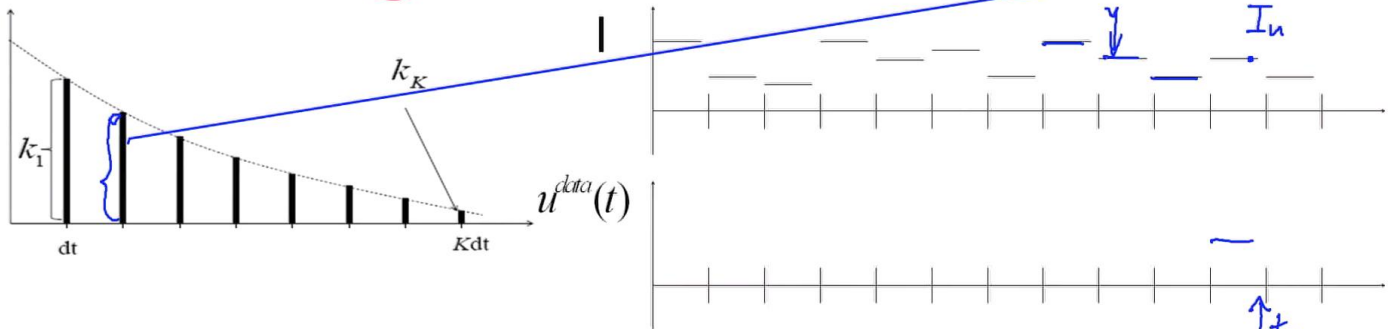
2m 47s

Neuronal Dynamics – 7.5 Parameter estimation: voltage

Linear in parameters = linear fit

$$u(t) = \int_0^\infty \kappa(s) I(t-s) ds + u_{rest}$$

$$u(t_n) = \sum k_k I_{n-k} + u_{rest}$$



$$E = \sum_n [u^{data}(t_n) - \sum_{k=1}^K k_k I_{n-k} - u_{rest}]^2$$

↩ / ↪ ↻

So here's my current. Now again, experimentalist inject a current, but in his set up, typically, it's a digital set up, so he injects current indiscrète time steps. One time step might be 0.1 milliseconds, something very short or a little bit longer, 1 millisecond. So the current is a discrete sequence of times. So this would be say my moment,, t that I'm interested in, t, and I have a current at this specific moment, let's call it t_n . I have a current in this time bin which is $i_{sub\ n}$. And now we measure data. The data might be like this in this time step. Now, the current was different in previous time steps. So what we do in this formula is, we take the current at time step n, n minus 1, and compare it and multiply it with a filter evaluated at 1, so k_1 times current n minus 1. And then we look at the current at n minus 2. So this one here would be n minus 2. And take the filter at this moment here, k_2 . This is how we calculate the voltage at time n of the model. Now, this is the model voltage. Here is the measured data voltage. What I would like to have is that the data voltage at time t_n is as close as possible to the model voltage. The model voltage is this term here.

Notes

Summary



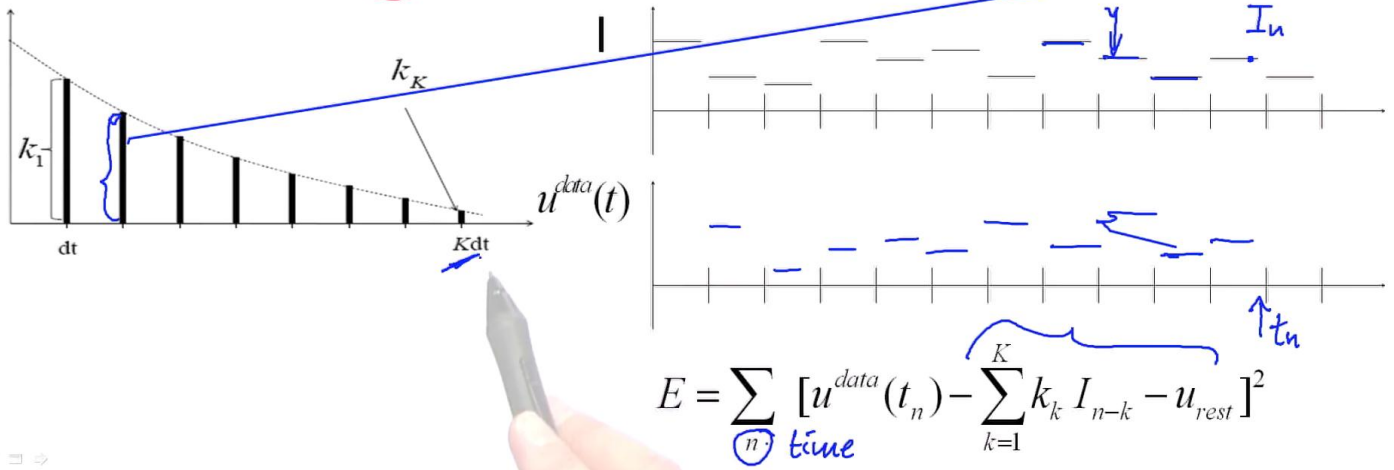
4m 12s

Neuronal Dynamics – 7.5 Parameter estimation: voltage

Linear in parameters = linear fit

$$u(t) = \int_0^\infty \kappa(s) I(t-s) ds + u_{rest}$$

$$u(t_n) = \sum k_k I_{n-k} + u_{rest}$$



Comes from over there. Now, this is true, this is my wish for t_n , for this time point here. But that should actually be true for all different time points. The data is different in each time step, and it should be true for all time steps. So this index n runs over all the different time steps. First index k runs from 1 to capital K . The capital K is the end of the filter. K times Δt is the duration of the filter.

Notes

Summary



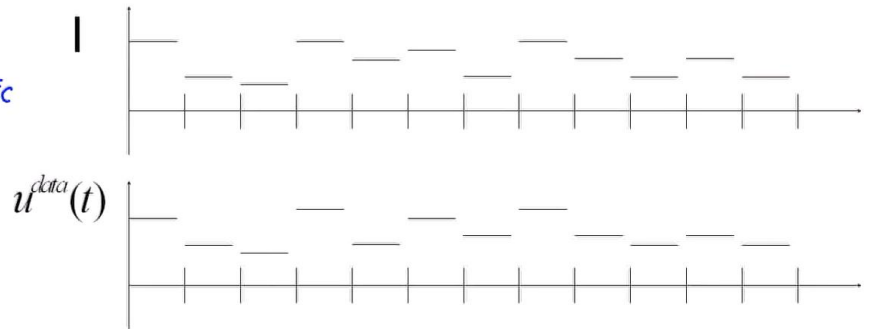
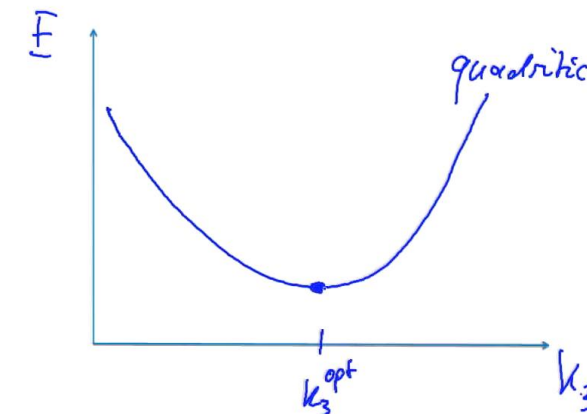
Neuronal Dynamics – 7.5 Parameter estimation: voltage

Linear in parameters = linear fit = quadratic optimization

Model $u(t) = \int_0^\infty \kappa(s) I(t-s) ds + u_{rest}$

$$u(t_n) = \sum_k k_k I_{n-k} + u_{rest}$$

Data $u^{data}(t)$



$$E = \sum_n [u^{data}(t_n) - \sum_{k=1}^K k_k I_{n-k} - u_{rest}]^2$$

(k1, k2, k3, ...)

Let's compare my data with my model. I sum up all time steps. e is supposed to be an error function. An error should be positive. That's the main reason why I square this. So the data voltage is larger than the model voltage or smaller than the model voltage. In both cases after squaring, I get a positive error. Now, let's plot this ever here, this, error as a function of one of the parameters, for example, the parameter k_3 . e is a function of k_1, k_2, k_3 , and so forth. So just take out one of these, say k_3 , and I plot e as a function of k_3 , and it will look like this. And then I will choose the value k_3 , the optimal value, k_3^{opt} is the one where this function is at a minimum. Now, since the voltage is linear in the parameters, the square of the voltage is quadratic in the parameters. In particular, it's a quadratic function of k_3 . Just take the square and multiply it out, and it will have terms of the form u^{data} squared in terms of the form u^{data} times k_k , say k_3 , and then you will have terms of the form k_3^2 in n minus 3 squared, which means k_3 is quadratic. So this function is quadratic in the parameters, and that means I can directly calculate the optimal value.

Notes

Summary



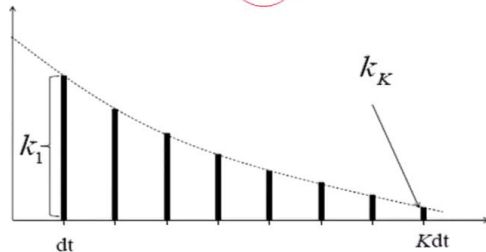
6m 55s

Neuronal Dynamics – 7.5 Parameter estimation: voltage

Linear in parameters = linear fit = quadratic

problem $u(t) = \int_0^\infty \kappa(s) I(t-s) ds + u_{rest}$

$$u(t_n) = \sum k_k I_{n-k} + u_{rest}$$



$$u(t_n) = \vec{k} \cdot \vec{x}_n$$

input time $\vec{x}$	x_1	x_2	x_3	...	x_K
$t=K+1$	I_K	I_{K-1}	I_{K-2}	...	I_1
$t=K+2$	I_{K+1}	I_K	I_{K-1}	...	I_2
$t=K+3$	I_{K+2}	I_{K+1}	I_K	...	I_3
...	...	...	...	...	...
$t=T$	I_T	I_{T-1}	I_{T-2}	...	I_{T-K+1}

dt

$$E = \sum [u^{data}(t_n) - \sum_{k=1}^K k_k I_{n-k} - u_{rest}]^2$$

Now, sometimes instead of writing these sums, it's more convenient to think of a vector notation. So here again, my current i of t , and this will be my value i at times n . And this will be my value i at times n minus 1, and maybe this one i at times n minus 1, 2, 3, 4. So let's take this summation as a vector product. I say I have this filter. My filter k is indeed a vector with entries k_1, k_2 , up to k lower index capital K , and I multiply this with a vector x_n . x_n would be my value i at times n , my current value at times n minus 1, my current value at times n minus k , and I made a mistake here. The vector starts at n minus 1 to be multiplied with k_1 . k_1 is multiplied with n minus 1. So with this notation, I get u of t_n is just my filter k as a vector times the vector x_n . Now, this was one of the vectors. x_n corresponds to the time point n . But we have many different time points in this sum. So for example, at time t equals k plus 1, I have the current at time k minus 1, k_2 , and so forth. At time k plus 3, I have the current at the time k plus 2, k plus 1, and so forth. Now, all these different entries are assembled in the lines.

Notes

Summary



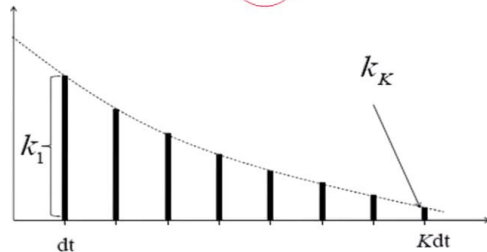
9m 01s

Neuronal Dynamics – 7.5 Parameter estimation: voltage

Linear in parameters = linear fit = quadratic

problem $u(t) = \int_0^\infty \kappa(s) I(t-s) ds + u_{rest} + \int_0^\infty \eta(s) S(t-s) ds$

$$u(t_n) = \sum k_k I_{n-k} + u_{rest}$$



$$u(t_n) = \vec{k} \cdot \vec{x}_n$$

input time $\vec{x}$	x_1	x_2	x_3	...	x_K
$t=K+1$	I_K	I_{K-1}	I_{K-2}	...	I_1
$t=K+2$	I_{K+1}	I_K	I_{K-1}	...	I_2
$t=K+3$	I_{K+2}	I_{K+1}	I_K	...	I_3
...	...	...	...	...	...
$t=T$	I_T	I_{T-1}	I_{T-2}	...	I_{T-K+1}

dt

$$E = \sum [u^{data}(t_n) - \sum_{k=1}^K k_k I_{n-k} - u_{rest}]^2$$

And then this way this whole formula becomes an exercise in optimization of a quadratic expression that includes matrices. So far I just treated the input current and the filter k , and I constructed this matrix of inputs. I said earlier that the experimentalist also observes the spike times. The η is not the linear filter. It can be treated just the same way. And we can have extra entries for the filter η and observed spike times in the past. So the total voltage it's just a generalized version of this one. Again, it remains linear in the parameters. It's linear in the parameters of the filter $kappa$. It's linear in the parameters of the filter η , and therefore, we have a quadratic problem.

Notes

Summary



11m 30s

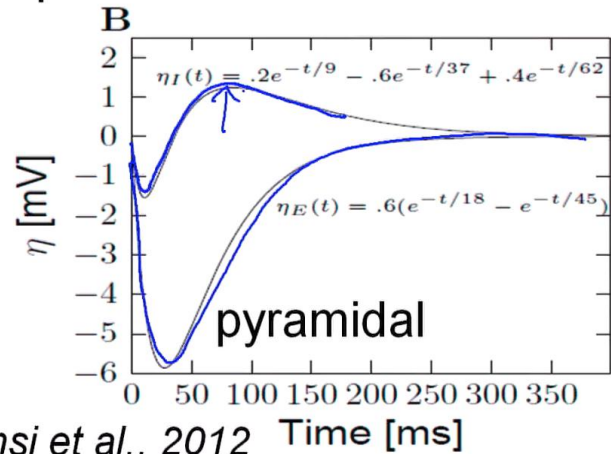
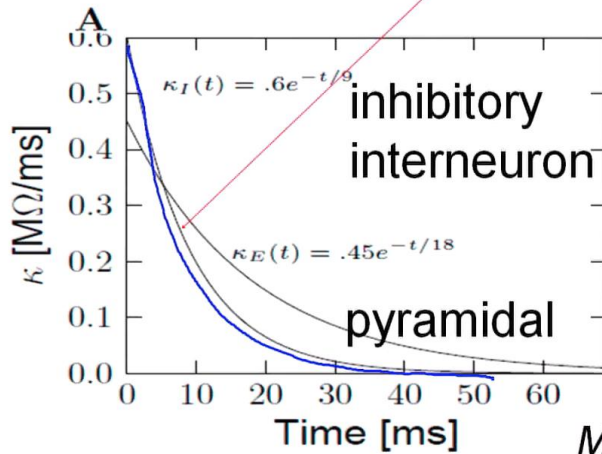
Neuronal Dynamics – 7.5 Extracted parameters: voltage

Subthreshold potential

$$u(t) = \int_0^{\infty} \kappa(s) I(t-s) ds + u_{rest} + \int \eta(s) S(t-s) ds$$

known input

known spike train



Mensi et al., 2012

✎ ✎ ✎ ✎

Now, if you apply this in order to optimize parameters of the sub threshold potential of a real neuron, let me find a filter kappa, which looks like this. And in fact, it can be fitted just by an exponential function. This would be the filter of an inhibitor neuron. Here's the filter of an excitatory neuron, again, an exponential filter with slightly different parameters. Now, if we extract the filter eta by this quadratic optimization procedure from the quadratic error function, then you would find a negative spike after potential for parametal neurons. There's inhibitor neurons, first a negative contribution, a high polarizing spike after potential followed by a depolarizing, a little bump, a depolarizing spike after potential.

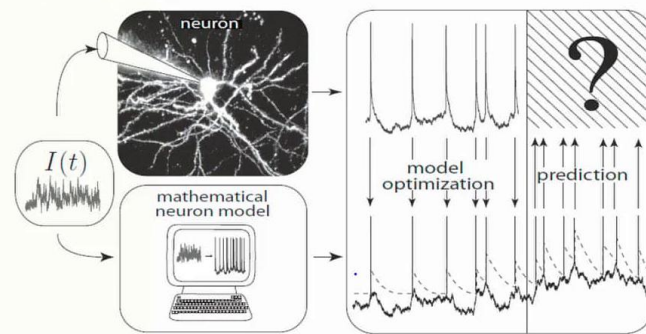
Notes

Summary



12m 31s

Neuronal Dynamics – What is a good neuron model?



- A) Predict spike times
- B) Predict subthreshold voltage
- C) Easy to interpret (not a 'black box')
- D) Flexible
- E) Systematic: 'optimize' parameters

That's what we have seen so far is with a GLM, with a spike responds motive with escape noise, we can optimize parameters in a systematic fashion. We can find the best prediction of the sub threshold voltage by solving a quadratic optimization problem, by finding the minimum of a quadratic error function, and therefore, straightforward procedures exist. But so far, our approach has been limited to the sub threshold membrane potential. Ideally, from a neuron model, we would also like to predict spikes, and that's next.

Notes

Summary



13m 36s