

Finding Tuning Curves for Point Neurons with Conductance-Based Synapses

Andreas Stöckel

October 19, 2017

Abstract

In the Neural Engineering Framework (NEF), individual neuron tuning curves are often characterized in terms of a maximum firing rate and an x -intercept. However, for LIF neurons with conductance-based synapses it is not immediately clear how maximum rate and x -intercept should be mapped to excitatory and inhibitory conductance input functions $g_E(x)$, $g_I(x)$. In this technical report we describe a method for deriving such functions and compare the resulting conductance-based tuning curves to current-based tuning curves with equivalent parameters. For large maximum rates and x -intercepts the conductance-based tuning curves possess a significantly steeper spike-rate onset compared to their current-based counterparts.

1 Introduction

In the case of a linear integrate and fire (LIF) neuron with current-based synapses, the Neural Engineering Framework (NEF) [1] defines the neural input current J given an input vector \vec{x} as an affine function

$$J(\langle \vec{x}, \vec{e} \rangle) = J(x) = \alpha x + J^{\text{bias}},$$

where \vec{e} is the encoding vector, α is the so called gain factor, and J^{bias} is a bias (or offset) current. The neuron response rate G over \vec{x} , the *tuning curve*, is then given as

$$G[J(\langle \vec{x}, \vec{e} \rangle)] = G[J(x)] = \frac{1}{\tau_{\text{ref}} + t_{\text{spike}}} = \begin{cases} \left(\tau_{\text{ref}} - \frac{\log\left(1 - \frac{v_{\text{th}}}{E_{\text{eq}}(J(x))}\right) \cdot C_m}{g_{\text{tot}}} \right)^{-1} & \text{if } E_{\text{eq}} > v_{\text{th}}, \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

where $g_{\text{tot}} = g_L$,

$$\text{and } E_{\text{eq}}(J(x)) = \frac{J(x)}{g_L} + E_L = \frac{\alpha x + J^{\text{bias}}}{g_L} + E_L.$$

Here, τ_{ref} is the refractory period, v_{th} is the neuron threshold potential, $E_{\text{eq}}(J)$ is the equilibrium potential given the current input, g_{tot} is the total conductance, and C_m is the membrane capacitance. For simplicity, let $v_{\text{th}} = 1$, $E_L = 0$, and $C_m = 1$. Given a target maximum rate G_{max} at $\langle \vec{x}, \vec{e} \rangle = x = 1$ and an x -intercept ξ (the input x for which $E_{\text{eq}} = v_{\text{th}}$), α and J^{bias} can be derived as a closed form expression. Let

$$\beta^{-1} = 1 - \exp(g_L(\tau_{\text{ref}} - G_{\text{max}}^{-1})), \quad \text{then } \alpha = \beta - J^{\text{bias}}, \quad \text{and } J^{\text{bias}} = \frac{\beta - 1}{1 - \xi}.$$

This technical report aims at providing a similar algorithm for LIF neurons with conductance-based synapses. We then compare equivalent current- and conductance-based tuning curves and provide an explanation for the observed steepness of the spike-rate onset.

2 Affine input functions for conductance-based synapses

Assuming constant, noise-free conductance values, the neuron response $G[g_E, g_I]$ for a LIF neuron with conductance-based synapses can be described in terms of eq. (1), where

$$g_{\text{tot}} = g_E + g_I + g_L, \quad \text{and } E_{\text{eq}}(g_E, g_I) = \frac{g_E E_E + g_I E_I + g_L E_L}{g_E + g_I + g_L}. \quad (2)$$

A one-dimensional tuning curve $G'[x]$ through the two-dimensional neuron response space is a parameterisation of g_E and g_I , $G'[x] = G[g_E(x), g_I(x)]$, where, in general, $g_E(x)$ and $g_I(x)$ are arbitrary non-negative functions on the interval $x \in [-1, 1]$.

Here, and similar to the current-based input translation function $J(x)$, we select $g_E(x)$ and $g_I(x)$ as affine functions (cf. fig. 1). Let $n > 0$ be the number of pre-synaptic ensembles projecting onto the neuron and define

$$g_E^n(x) = ax + b' = ax + \frac{b}{n}, \quad g_I^n(x) = cx + d' = cx + \frac{d}{n}, \quad (3)$$

with constraints $b > |a| \cdot n$, $d > |c| \cdot n$ to ensure non-negativity. Note that the parameter n allows additive superposition of the value decoded from multiple pre-synaptic ensembles

$$G'[x_1 + x_2] = G[g_E^1(x_1 + x_2), g_I^1(x_1 + x_2)] = G[g_E^2(x_1) + g_E^2(x_2), g_I^2(x_1) + g_I^2(x_2)].$$

A parameter such as n is not required in standard NEF neuron ensembles with current-based synapses. Additive superposition of pre-synaptic values is implemented by declaring the bias current J^{bias} to be a part of the neuron model. In this case, the input translation function $J(x) = \alpha x$ is just a linear function and intrinsically additive. In contrast, the translation functions for conductance-based synapses as proposed in eq. (3) must decode the

excitatory and inhibitory biases b' , c' from the pre-synaptic populations.¹ Conductance biases are not part of the neuron model, since the non-negativity constraint would be violated: without the bias terms b' , d' , the decoded functions $g_E^n(x)$, $g_I^n(x)$ would not be non-negative!

Given the conductance translation functions in eq. (3), we now have to select parameters a , b , c , d with respect to an x -intercept ξ and a maximum firing rate G_{\max} . As in the introduction, we assume without loss of generality $C_m = 1$, $v_{\text{th}} = 1$, $E_L = 0$.

First, we solve for the x -intercept. In the case of conductance-based synapses, we need to find pairs of g_E , g_I at which the neuron just starts spiking. Equating E_{eq} with $v_{\text{th}} = 1$ and solving for g_E gives

$$E_{\text{eq}} = \frac{g_E E_E + g_I E_I}{g_L + g_E + g_I} \stackrel{!}{=} v_{\text{th}} = 1 \Leftrightarrow g_E = \frac{g_L - g_I(E_I - 1)}{(E_E - 1)},$$

which is a linear in g_I . Combining with the conductance values at the x -intercept ξ , $g_E(\xi)$ and $g_I(\xi)$ (with $n = 1$), yields

$$a\xi + b' = \frac{g_L - (c\xi + d')(E_I - 1)}{(E_E - 1)}. \quad (4)$$

Similarly, we can solve eqs. (1) and (2) for $G[g_E, g_I] = G_{\max}$

$$G[g_E, g_I] = \frac{1}{\tau_{\text{ref}} + t_{\text{spike}}} \stackrel{!}{=} G_{\max} \Leftrightarrow t_{\text{spike}} = \frac{1}{G_{\max}} - \tau_{\text{ref}},$$

$$t_{\text{spike}} \stackrel{!}{=} -\frac{\log\left(1 - \frac{g_L + g_E + g_I}{g_E E_E + g_I E_I}\right)}{g_L + g_E + g_I} \Rightarrow \exp(-t_{\text{spike}} \cdot (g_L + g_E + g_I)) = 1 - \frac{g_L + g_E + g_I}{g_E E_E + g_I E_I}.$$

Unfortunately, this equation has no closed form solution when solving for g_E . Instead, we must numerically find the root of

$$f(g_E) = (g_E E_E + g_I E_I) \cdot (\exp(-t_{\text{spike}}(g_L + g_E + g_I)) - 1) + g_L + g_E + g_I. \quad (5)$$

This can be easily accomplished with Newton's method. The corresponding derivative is

$$f'(g_E) = (E_E - t_{\text{spike}} \cdot (g_E E_E + g_I E_I)) \cdot \exp(-t_{\text{spike}} \cdot (g_L + g_E + g_I)) - E_E + 1. \quad (6)$$

Newton's method usually converges to the solution within a few iterations when initialized with large g_E .

We have solved the firing rate equation for the x -intercept and a maximum rate G_{\max} .

¹Of course, the same technique is applicable to current-based synapses as well, where J^{bias} can be decoded from the pre-populations.

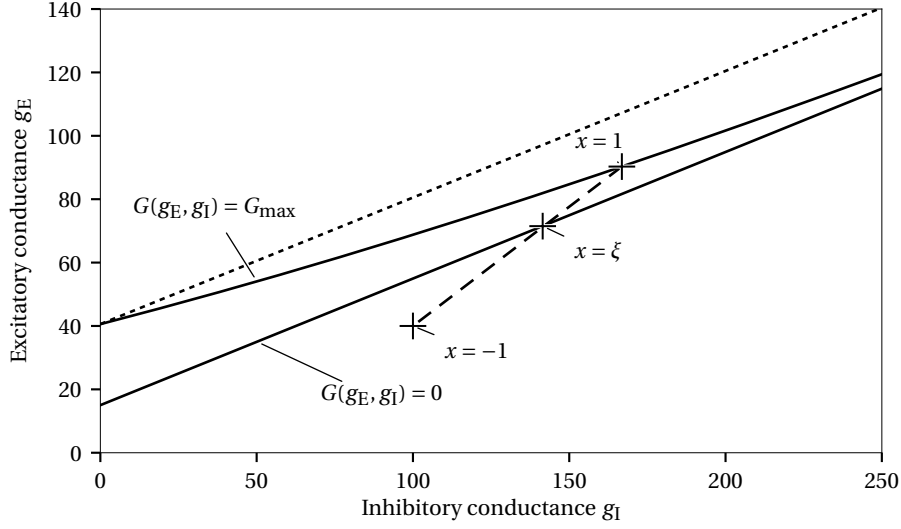


Figure 1: Two-dimensional view of the tuning curve (dashed line) within the g_E , g_I space. Solid lines correspond to contour lines of the neuron response curve. The bottom line corresponds to points at which the neuron response is just zero, that is $E_{\text{eq}} = v_{\text{th}}$, the top line corresponds to the line at which the desired maximum rate G_{max} is reached. The dotted line corresponds to the approximation of the G_{max} contour line as a line parallel to the x -intercept contour and offset by a g_E^{offs} chosen such that the line passes through the true maximum rate contour at $g_I = 0$.

On their own, these solutions are not really useful when solving for the affine function parameters a , b , c , d as they require a numerical solution of eq. (5). Solving for both x -intercept and maximum rate G_{max} must thus be performed as part of a numerical optimization scheme. To this end, we approximate the g_E at which G_{max} is reached as a line parallel to the x -intercept equation (4) shifted by an offset g_E^{offs} (fig. 1)

$$g_E(1) = a + b' = \frac{g_L - g_I(1) \cdot (E_I - 1)}{(E_E - 1)} + g_E^{\text{offs}} = \frac{g_L - (c + d') \cdot (E_I - 1)}{(E_E - 1)} + g_E^{\text{offs}}. \quad (7)$$

The offset g_E^{offs} can be obtained by numerically solving $G[g_E(1), g_I(1)] = G_{\text{max}}$ for $g_I(1)$ using Newton's methods (eqs. (5) and (6)), where $g_I(1)$ is a guess for the inhibitory conductance remaining when the neuron spikes with the maximum rate G_{max} . Initially, a valid assumption is $g_I(1) = 0$, i.e. there is no inhibitory conductance when the neuron spikes with the maximum spike rate.

We now have a system of linear equations, namely eqs. (4) and (7), with four unknowns and constraints $b > |a| \cdot n$, $d > |c| \cdot n$ for non-negativity of $g_E^n(x)$, $g_I^n(x)$ on $[-1, 1]$. Since we solve for four variables and have two input values (maximum rate G_{max} and x -intercept), we need to eliminate two degrees of freedom for a unique solution. The non-negativity constraint eliminates one degree of freedom, leaving one further constraint. A natural

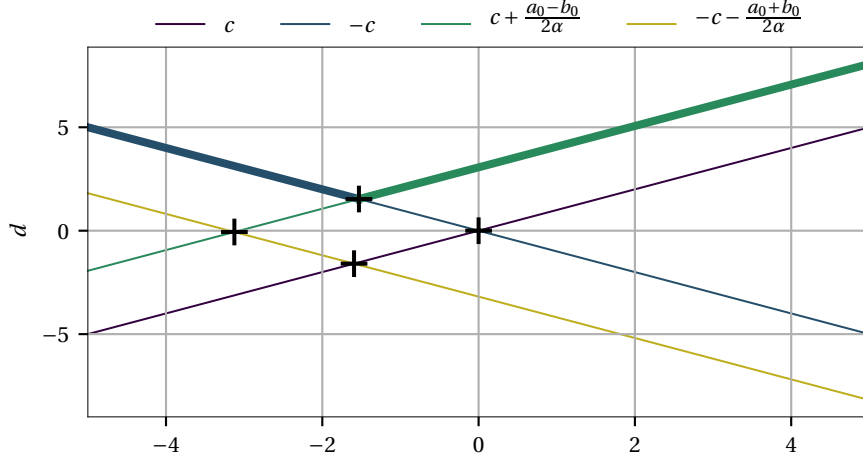


Figure 2: Sketch showing the the parameter d as a function of c (thick line) given in eq. (11). This function is the maximum of four affine functions with slope -1 and 1 . The minimum of this function can be found by calculating the intersection points of the four functions (crosses) and choosing the intersection point with the largest value for d .

objective is energy minimization, which, in this context, corresponds to minimizing the overall conductance in the system, i.e. the number of open ion channels

$$\text{minimize } \int_{-1}^1 g_E(x) + g_I(x) dx = 2(b + d). \quad (8)$$

Solving eqs. (4) and (7) for a, b yields

$$a = \alpha c + a_0, \quad b = \alpha d + b_0, \quad (9)$$

where $\alpha = -\frac{E_I - 1}{E_E - 1}$, $a_0 = -\frac{g_E^{\text{offs}}}{\xi - 1}$, $b_0 = \frac{g_E^{\text{offs}} \cdot \xi \cdot (E_E - 1) + g_L \cdot (\xi - 1)}{(E_E - 1) \cdot (\xi - 1)}$.

Note that the slope α in the above equation is always positive for $E_I < 1$ and $E_E > 1$. This condition is always true for sane neural parameter sets where the inhibitory channel *inhibits* spike production and the excitatory *excites* the neuron to spike.

Given the energy minimization constraint eq. (8) we can now rewrite the entire problem as minimization of the parameter d

$$\begin{aligned} \text{minimize } b + d &= d(\alpha + 1) + b_0 \stackrel{\alpha > 0}{\Leftrightarrow} \text{minimize } d \\ \text{w.r.t } d &\geq |c| \cdot n \Leftrightarrow d \geq cn \wedge d \geq -cn, \\ b &\geq |a| \cdot n \Leftrightarrow d \geq cn + \frac{a_0 n - b_0}{\alpha} \wedge d \geq -cn - \frac{a_0 n + b_0}{\alpha}. \end{aligned} \quad (10)$$

In order to satisfy the minimization problem eq. (10) the parameter d must be the mini-

imum possible value which satisfies all constraints, or, put differently, we need to minimize d over c and a set of greater-or-equal constraints

$$d_{\min} = \min_c \max \left\{ cn, -cn, cn + \frac{a_0 n - b_0}{\alpha}, -cn - \frac{a_0 n + b_0}{\alpha} \right\}.$$

Geometrically, the minimum must be located on one of the intersection point of these four affine functions (fig. 2). Since two pairs of the individual functions are parallel (possess the same slope), there are four intersection points and the solution is given as the (c, d) -tuple from the below set with the maximum value for d

$$\left\{ (0, 0), \left(-\frac{a_0 n + b_0}{2\alpha n}, -\frac{a_0 n + b_0}{2\alpha} \right), \left(-\frac{a_0 n - b_0}{2\alpha n}, \frac{a_0 n - b_0}{2\alpha} \right), \left(-\frac{a_0}{\alpha}, -\frac{b_0}{\alpha} \right) \right\}. \quad (11)$$

Values for a and b can be computed using the linear equations in eq. (9). We can now calculate $g_I(1) = c + d$, numerically solve for the $g_E(1)$ at which the maximum rate is reached, and derive a new g_E^{offs} such that the maximum rate target function passes through the point $(g_E(1), g_I(1))$. The entire process is described below in algorithm 1.

Algorithm 1 Algorithm computing the affine function parameters a, b, c, d .

```

err ← ∞
gIoffs ← 0
while err > errmax do
  gEmax ← gE for which G[gE, gIoffs] = Gmax
  gEoffs ← gEmax -  $\frac{g_I - g_I(E_I - 1)}{E_E - 1}$ 
  Compute  $a, a_0, b_0$  (cf. eq. 9)
  Select  $c, d$  with maximum  $d$  (cf. eq. 11)
  Compute  $a, b$  (cf. eq. 9)
  err ← |Gmax - G[a + b, c + d]|
  gIoffs ← c + d
end while
return a, b, c, d

```

3 Current- and conductance-based tuning curve comparison

Figure 3 shows a comparison between the tuning curves of a LIF neuron with either current- or conductance-based synapses for a set of maximum rates and x -intercepts. The neuron has a membrane time constant $\tau_{RC} = 20$ ms and a refractory period of $\tau_{\text{ref}} = 2$ ms. Especially for larger x -intercepts and maximum rates G_{max} , the spike onset is significantly steeper for the conductance-based synapses, i.e. after being completely silent, the neuron “jumps” to a relatively high firing rate.

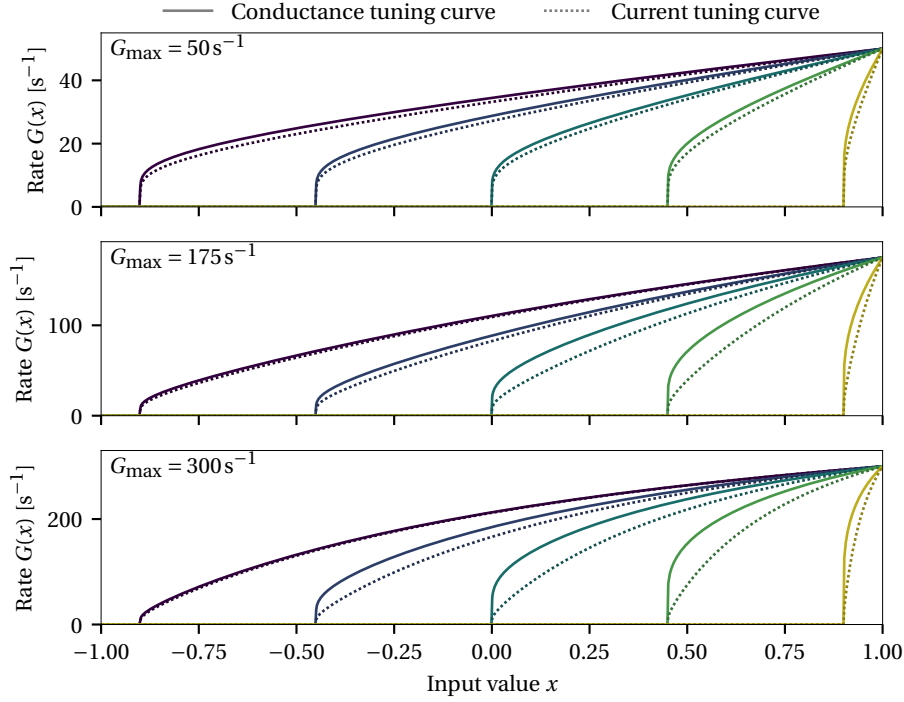
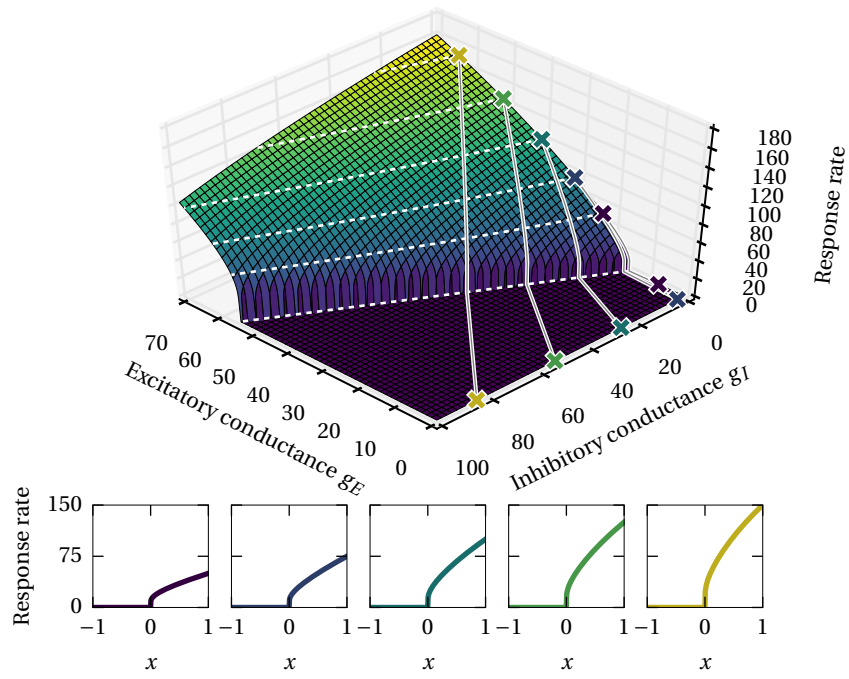


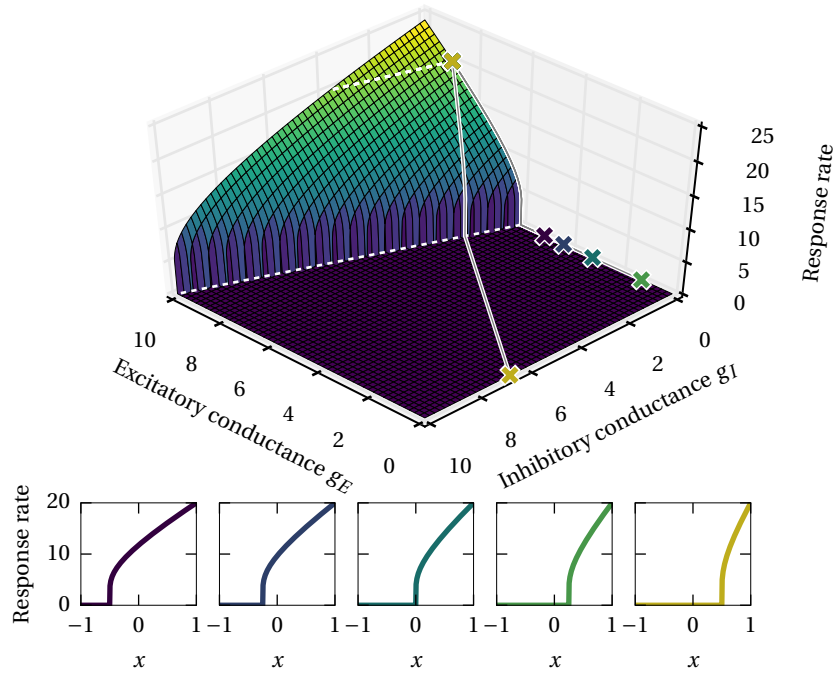
Figure 3: Comparison between tuning curves for a LIF neuron with conductance- and current-based synapses for a set of maximum rates and x -intercepts. Tuning curves are derived using the methods described in this report. Solid lines correspond to the tuning curves of the neurons with conductance-based synapses, dotted lines correspond to the tuning curves of the neurons with current-based synapses.

To explain this behaviour, we first have to consider eqs. (1) and (2), especially the term g_{tot} , which is the total conductance in the system and directly determines the convergence rate of the membrane potential towards the equilibrium potential E_{eq} . Thus, as long as the equilibrium potential is smaller than the threshold potential ($E_{\text{eq}} < v_{\text{th}}$), even if $g_{\text{tot}} \rightarrow \infty$, the output spike rate will be zero. However, if $g_{\text{tot}} \rightarrow \infty$ and the equilibrium potential reaches v_{th} , the neuron will spike with the maximum possible rate, which is τ_{ref}^{-1} . In other words, a large g_{tot} results in a steep spike-rate onset.

Now, why do a large maximum rate and x -intercept result in a large g_{tot} , and thus the steepness observed in fig. 3? This question is best answered with figs. 4 and 5, which depict the conductance-based tuning curves derived with the above algorithm embedded into the underlying neuron response curve surface. In particular, Figure 4a depicts tuning curves with increasing maximum rate and an x -intercept $\xi = 0$, which can be geometrically interpreted as the spike onset being located on the centre of the tuning curve. To minimize the total conductance, the parameter selection algorithm selects $g_{\text{I}}(x) = 0$ as long as possible, however, if $g_{\text{I}}(x)$ were to stay zero for larger G_{max} , the excitatory conductance

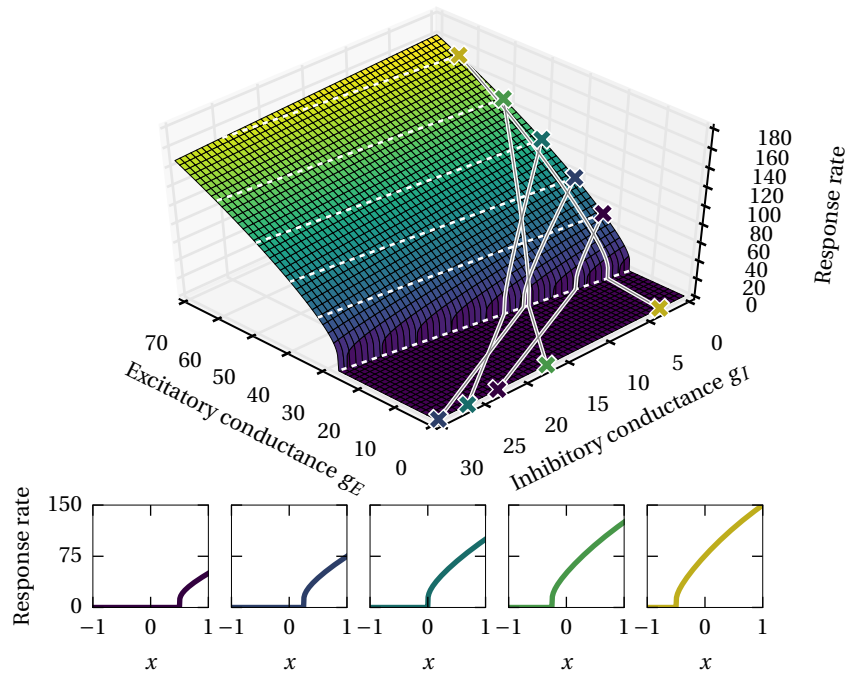


(a) Variable maximum rate

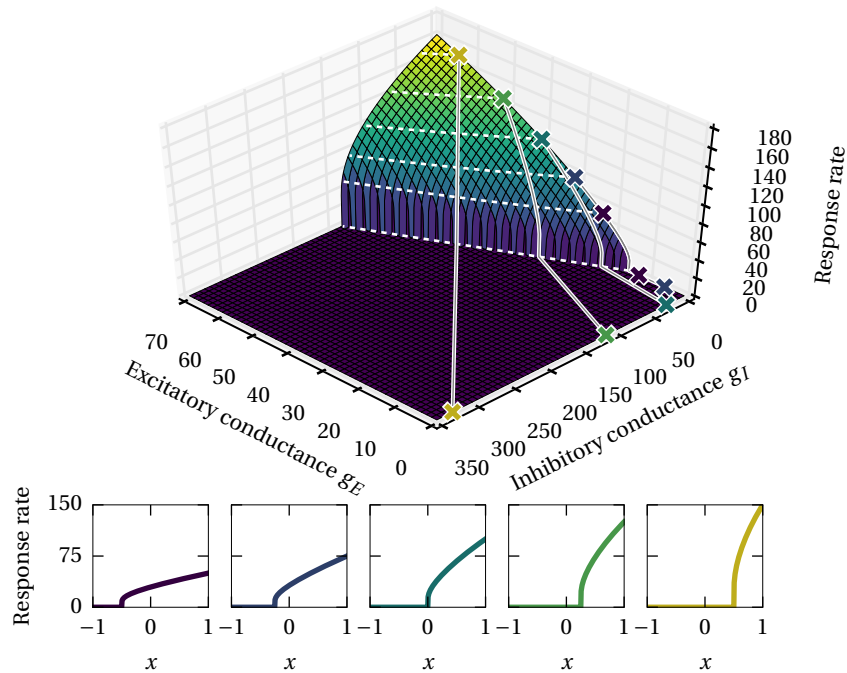


(b) Variable x -intercept

Figure 4: Examples of tuning curves $G[x]$ embedded into the 2D neuron response space. Tuning curves in the bottom part of the figure correspond to the curve $(g_E(x), g_I(x))$ with same-coloured crosses in the top part of the figure.



(a) Decreasing x -intercept, increasing maximum rate



(b) Increasing x -intercept, increasing maximum rate

Figure 5: Examples of tuning curves $G[x]$ embedded into the 2D neuron response space. Tuning curves in the bottom part of the figure correspond to the curve $(g_E(x), g_I(x))$ with same-coloured crosses in the top part of the figure.

would, illegally, extend into negative regions. Therefore, and as visible in Figure 4a the algorithm selects a non-zero inhibitory conductance, and the tuning-curve diagonally crosses the non-zero spike rate boundary. As depicted in Figure 4b, the same is true for increasing x -intercepts at constant maximum spike rates. The non-zero inhibitory conductance results in an overall increased total conductance at the point of the x -intercept and thus in the observed steepness of the spike onset.

4 Conclusion

Given an x -intercept and a maximum rate G_{\max} , the method presented in section 2 computes parameters a , b , c , d for affine input translation functions $g_E(x)$, $g_I(x)$ in conjunction with conductance-based synapses in the NEF. The algorithm guarantees a unique solution by demanding non-negativity of g_E and g_I and minimizing the average conductance in the system. The resulting translation functions $g_E(x)$, $g_I(x)$ define the tuning curve of the neuron, which in turn is essential when solving for synaptic weights.

As mentioned above, the particular choice of input translation functions is arbitrary. In fact, when defining the tuning curve of a neuron, any translation function $g_E(x)$, $g_I(x)$ can be used, under the condition that these functions can be decoded from the pre-population. For a network of conductance-based neurons, decoding is restricted to non-negative weight matrices; it must be possible to represent $g_E(x)$ and $g_I(x)$ as a non-negative linear combination of the tuning curves in the pre-population, which restricts possible input translation functions and transformations between ensembles. In practice, it will only be possible to non-negatively decode linear and (to a lesser degree) quadratic functions.

To decode arbitrary functions from ensembles with conductance-based synapses, it is necessary to let go of the concept of translation functions. The translation function $J(x)$ makes sense in the standard NEF because there is a one-to-one mapping between currents and non-zero output rates. For conductance-based synapses the input translation functions map non-zero output rates to a single tuple (g_E, g_I) although the underlying mapping is not one-to-one. Collapsing the two-dimensional conductance space onto a line eliminates a degree of freedom required for arbitrary non-negative decodings.

Still, while the method above does not solve the problem of computing arbitrary functions between ensembles of neurons with conductance-based synapses, it can be used to derive tuning curves for neurons with conductance-based synapses which have a simple (affine) shape in the conductance space. In contrast, matching conductance translation functions to standard LIF tuning curves would result in a more complex non-linear/affine pair of translation functions.

References

- [1] Chris Eliasmith and Charles H. Anderson. *Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*. Cambridge, Massachusetts: MIT press, 2003. 380 pp.