Appendix A1

Mathematical analysis of IntFire4

The `IntFire4` mechanism is an artificial spiking cell with a fast, monoexponentially decaying excitatory current e and a slower biexponential (similar to alpha function) inhibitory current i_2 that are summed by an even slower leaky integrator. It fires when the membrane state m reaches 1; after firing, only the membrane state returns to 0. The dynamics of `IntFire4` are specified by four time constants--τ_e for the excitatory current, τ_{i_1} and τ_{i_2} for the inhibitory current, and τ_m for the leaky integrator--and it is assumed that $\tau_e < \tau_{i_1} < \tau_{i_2} < \tau_m$. However, the differential equations that govern `IntFire4` are more conveniently written in terms of rate constants, i.e.

$$\frac{de}{dt} = -k_e e \qquad \text{Eq. A1.1}$$

$$\frac{di_1}{dt} = -k_{i_1} i_1 \qquad \text{Eq. A1.2}$$

$$\frac{di_2}{dt} = -k_{i_2} i_2 + a_{i_1} i_1 \qquad \text{Eq. A1.3}$$

$$\frac{dm}{dt} = -k_m m + a_e e + a_{i_2} i_2 \qquad \text{Eq. A1.4}$$

where each rate constant k is the reciprocal of the corresponding time constant, and $k_e > k_{i_1} > k_{i_2} > k_m$. An input event adds its weight w instantaneously to e or i_1, depending on whether w is > 0 (excitatory) or < 0 (inhibitory), respectively. The states e, i_1, i_2, and m are normalized by the constants a_e, a_{i_1}, and a_{i_2}, so that an excitatory weight w_e drives e and m to a maximum of w_e, and an inhibitory weight w_i drives i_1, i_2, and m to a minimum of w_i (see Fig. 10.15).

This system of equations can be solved by repeatedly making use of the fact that the solution to

$$\frac{dy}{dt} = -k_1 y + a\,\mathrm{e}^{-k_2 t} \qquad \text{Eq. A1.5}$$

is

$$y = y_0 e^{-k_1 t} + b(e^{-k_1 t} - e^{-k_2 t}) \qquad \text{Eq. A1.6}$$

where y_0 is the value of y at $t = 0$, and $b = a / (k_2 - k_1)$. Note that when $a > 0$ and $k_2 > k_1$, we may conclude that $b > 0$.

The solution to Eqns. A1.1-A1.4 is

$$e(t) = e_0 e^{-k_e (t - t_0)} \qquad \text{Eq. A1.7}$$

$$i_1(t) = i_{1_0} e^{-k_{i_1}(t - t_0)} \qquad \text{Eq. A1.8}$$

$$i_2(t) = i_{2_0} e^{-k_{i_2}(t - t_0)} + b_{i_1} i_1 (e^{-k_{i_2}(t - t_0)} - e^{-k_{i_1}(t - t_0)}) \qquad \text{Eq. A1.9}$$

$$\begin{aligned} m(t) = {} & m_0 e^{-k_m (t - t_0)} \\ & + b_e (e^{-k_m (t - t_0)} - e^{-k_e (t - t_0)})\, e_0 \\ & + b_{i_2} (e^{-k_m (t - t_0)} - e^{-k_{i_2}(t - t_0)})\, i_{2_0} \\ & + b_{i_2} b_{i_1} (e^{-k_m (t - t_0)} - e^{-k_{i_2}(t - t_0)})\, i_{1_0} \\ & - b_{i_2} b_{i_1} \frac{k_{i_2} - k_m}{k_{i_1} - k_m} (e^{-k_m (t - t_0)} - e^{-k_{i_1}(t - t_0)})\, i_{1_0} \end{aligned} \qquad \text{Eq. A1.10}$$

where

t_0 is the time of the most recent input event

e_0, i_{1_0}, i_{2_0}, and m_0 are the values of e , i_1, i_2, and m immediately after that event was handled

IntFire4 uses self-events to successively approximate the firing time. At initialization, a self-event is issued that will return at $t = 10^9$ ms (i.e. never). Arrival of a new event at time t_{event} causes the following sequence of actions:

- The current values of the states e , i_1, i_2, and m are calculated analytically from Eqns. A1.7-A1.10.
- The values of e_0, i_{1_0}, i_{2_0}, and m_0 are updated to the current values of e , i_1, i_2, and m, and the value of t_0 is updated to t_{event}.
- If $m > 1 - \epsilon$, the cell fires and m is reset to 0.

- If the event was a self-event, the next firing time is estimated and a new self-event is issued that will return at that time.
- If the event was an input event, then depending on whether it was excitatory or inhibitory (i.e. weight $w < 0$ or > 0), w is instantaneously added to e or i_1, respectively. That done, the next firing time is estimated, and the yet-outstanding self-event is moved to that time.

The next firing time is approximated from the values of m and its derivative immediately after the event is handled. If $m(t_0)' \leq 0$, then the estimated firing time is set to 10^9, i.e. never. If $m(t_0)' > 0$, the estimated firing time is $(1-m(t_0))/m(t_0)'$. In the following sections we prove that this strategy produces an estimate that is never later than the true firing time; otherwise, the simulation would be in error.

From a practical perspective, it is also important that successive approximations converge rapidly to the true firing time, to avoid the overhead of a large number of self events. Since the slope approximation is equivalent to Newton's method for finding the t at which $m = 1$, we only expect slow convergence when the maximum value of m is close to 1. Using a sequence of self-events is superior to carrying out a complete Newton method solution for the firing time, because it is most likely that external (input) events will arrive in the interval between firing times, invalidating the computation of the next firing time. The number of iterations that should be carried out per self-event remains an experimental question, because self-event overhead depends partly on the number of outstanding events in the event queue. A single Newton iteration generally takes longer than the overhead associated with self-events.

Proof that the estimate is never later than the true firing time

For notational clarity, we will use m_0 and m_0' to refer to the values of m and m' immediately after the event is handled. The proof consists of two major parts. First we show that if $m_0' \leq 0$, then m(t) remains < 1. Then we show that if $m_0' > 0$, then $(1-m_0)/m_0'$ underestimates the firing time. This latter part is divided into the cases $m_0 \leq 0$, and $m_0 > 0$. First, however, we present a useful lemma.

Lemma:

If

$$k_1 > k_2 > k \qquad \text{Eq. A1.11}$$

$$f_1(t) = e^{-k\,t} - e^{-k_1\,t} \qquad \text{Eq. A1.12}$$

$$f_2(t) = e^{-k\,t} - e^{-k_2\,t} \qquad \text{Eq. A1.13}$$

then

$$\frac{f_1(t)}{k_1 - k} \le \frac{f_2(t)}{k_2 - k} \qquad \text{Eq. A1.14}$$

for all $t \ge 0$.

Proof:

First note that $f_1(0) = f_2(0) = 0$ so the lemma holds at $t = 0$. Also note that $f_1{}'(0) = k_1 - k$ and $f_2{}'(0) = k_2 - k$ so both sides of the inequality we are trying to prove have slope 1 at $t = 0$.

Next consider $t > 0$. $e^{-k\,t} > e^{-k_1\,t} > e^{-k_2\,t}$ so it is safe to divide by $e^{-k\,t} - e^{-k_2\,t}$, and we can write

$$\frac{f_1(t)}{k_1 - k} - \frac{f_2(t)}{k_2 - k} = \frac{k_2 - k}{f_2(t)} \left(\frac{f_1(t)}{f_2(t)} \, \frac{k_2 - k}{k_1 - k} - 1 \right) \qquad \text{Eq. A1.15}$$

Analyzing the right hand side of this equation, we see that the ratio $(k_2 - k) \,/\, f_2(t)$ is positive. Also, $(k_2 - k)/(k_1 - k) < 1$. Furthermore, f_1 and f_2 are positive, and since $e^{-k\,t} - e^{-k_2\,t}$ then $f_1 \,/\, f_2 < 1$. Thus the expression inside the parentheses is negative, and the entire right hand side of Eq. A1.16 is < 0. This completes the proof of the lemma.

Note that Eq. A1.15 can be expressed as

$$f_1(t) \, \frac{k_2 - k}{k_1 - k} \le f_2(t) \qquad \text{Eq. A1.16}$$

Also, in the limit as k_2 approaches k, we have

$$\frac{f_1(t)}{k_1 - k} \le t\, e^{-k\,t} \qquad \text{Eq. A1.17}$$

Part 1: if $m_0{}' \le 0$, then $m(t)$ remains < 1

We now prove that if $m_0{}' \le 0$, then m(t) remains < 1 (i.e. the firing time is infinity) regardless of e, i_1, or i_2. Since we are trying to predict the trajectory of m based on the values of m and m' immediately following the most recent event, it will be advantageous to think in terms of the time that has elapsed since that event, i.e. relative time, rather than absolute time. Therefore we substitute t for t - t_0, and rewrite Eq. A1.10 as

$$\begin{aligned} m(t) = {} & m_0 e^{-k_m t} \\ & + b_e (e^{-k_m t} - e^{-k_e t})\, e_0 \\ & + b_{i_2} (e^{-k_m t} - e^{-k_{i_2} t})\, i_{2_0} \\ & + b_{i_2} b_{i_1} (e^{-k_m t} - e^{-k_{i_2} t})\, i_{1_0} \\ & - b_{i_2} b_{i_1} \frac{k_{i_2} - k_m}{k_{i_1} - k_m} (e^{-k_m t} - e^{-k_{i_1} t})\, i_{1_0} \end{aligned} \qquad \text{Eq. A1.18}$$

From the lemma we see that the sum of the last two major terms on the right hand side is ≤ 0. Factoring out the common multiplier $b_{i_2} b_{i_1} i_{1_0}$ from these terms leaves the expression

$$(e^{-k_m t} - e^{-k_{i_2} t}) - \frac{k_{i_2} - k_m}{k_{i_1} - k_m} (e^{-k_m t} - e^{-k_{i_1} t})$$

which is positive because $k_{i_1} > k_{i_2} > k_m$. However, the multiplier $b_{i_2} b_{i_1} i_{1_0}$ itself is ≤ 0 because i_{1_0} is ≤ 0 while b_{i_1} and b_{i_2} are both > 0.

Thus

$$\begin{aligned} m(t) \le {} & m_0 e^{-k_m t} \\ & + b_e (e^{-k_m t} - e^{-k_e t})\, e_0 \\ & + b_{i_2} (e^{-k_m t} - e^{-k_{i_2} t})\, i_{2_0} \end{aligned} \qquad \text{Eq. A1.19}$$

The last term here is negative (except at $t = t_0$, where it is 0), and we can use our lemma again to replace it with something that is not so negative, i.e.

$$\begin{aligned} m(t) \le {} & m_0 e^{-k_m t} \\ & + b_e (e^{-k_m t} - e^{-k_e t})\, e_0 \\ & + b_{i_2} \frac{k_{i_2} - k_m}{k_e - k_m} (e^{-k_m t} - e^{-k_e t})\, i_{2_0} \end{aligned} \qquad \text{Eq. A1.20}$$

Rewriting this as

$$m(t) \le m_0 e^{-k_m t} + \frac{1}{k_e - k_m} (e^{-k_m t} - e^{-k_e t}) (a_e e_0 + a_{i_2} i_{2_0}) \qquad \text{Eq. A1.21}$$

we note that $a_e\, e_0 + a_{i_2}\, i_{2_0}$ is $m_0' + k_m\, m_0$, so

$$m(t) \le m_0 e^{-k_m t} + \frac{k_m}{k_e - k_m} (e^{-k_m t} - e^{-k_e t})\, m_0 + \frac{1}{k_e - k_m} (e^{-k_m t} - e^{-k_e t}) m_0' \qquad \text{Eq. A1.22}$$

We have stipulated that $m_0' \le 0$, so the last term is ≤ 0 and we can remove it and write

$$m(t) \le m_0 \left[e^{-k_m t} + \frac{k_m}{k_e - k_m} (e^{-k_m t} - e^{-k_e t}) \right] \qquad \text{Eq. A1.23}$$

Since $m_0 < 1$, we only have to prove that the bracketed expression is ≤ 1. Clearly it is 1 when $t = 0$. Factoring this expression gives

$$\frac{k_e}{k_e - k_m} e^{-k_m t} - \frac{k_m}{k_e - k_m} e^{-k_e t}$$

whose derivative is

$$-\frac{k_e\, k_m}{k_e - k_m} e^{-k_m t} + \frac{k_e\, k_m}{k_e - k_m} e^{-k_e t}$$

or

$$-\frac{k_e\, k_m}{k_e - k_m} (e^{-k_m t} - e^{-k_e t})$$

which is 0 at $t = 0$ and negative for $t > 0$. A function that is 1 at $t = 0$ and has a negative derivative for $t > 0$ must be ≤ 1 for $t > 0$.

This completes Part 1 of the proof. Next we prove that, if $m_0' > 0$, the first Newton iteration estimate $(1 - m_0) / m_0'$ is never later than the true firing time.

Part 2: if *m′* > 0, 1 - *m* / *m′* underestimates the firing time

The last thing to do is to prove that, if $m_0' > 0$, the Newton iteration $(1 - m_0) / m_0'$ is never later than the firing time. We start from Eq. A1.22, but since we now stipulate that

$m_0' > 0$, the last term is positive. According to our lemma, we can replace it with the larger term $t\,e^{-k_m t}\,m_0'$ to get

$$m(t) \le m_0 e^{-k_m t} + \frac{k_m}{k_e - k_m}\left(e^{-k_m t} - e^{-k_e t}\right) m_0 + t\,e^{-k_m t}\,m_0' \qquad \text{Eq. A1.24}$$

Consider the case where $m_0 > 0$. The sum of the first two terms is $\le m_0$ and the third term is $\le t\, m_0'$, so

$$m(t) \le m_0 + m_0'\, t \qquad \text{Eq. A1.25}$$

and the Newton iteration underestimates the firing time.

Now consider case where $m_0 < 0$. The second term of Eq. A1.24 is ≤ 0 so we can throw it out and write

$$m(t) \le (m_0 + m_0'\, t)e^{-k_m t} \qquad \text{Eq. A1.26}$$

We complete our proof by applying a geometric interpretation to this inequality. The value of t at which the line $y(t) = m_0 + m_0'\, t$ intersects $y = 1$ is the estimated firing time found by a Newton iteration. Equation A1.24 shows that the trajectory of the membrane state variable runs at or below that line. Consequently, the Newton iteration underestimates the true firing time.

