The quadratic integrate-and-fire neuron

By: Kamal Abu-Hassan

1. Introduction

This document provides a description of the quadratic integrate-and-fire neuron model, its mathematical representation, computational implementation with MATLAB software and an example simulation of the model.

2. The model

An alternative to the leaky integrate-and-fire neuron (LIF) is the quadratic I&F neuron, also known as the theta-neuron (Ermentrout, 1996; Ermentrout & Kopell, 1986; Izhikevich, 2004). This model is canonical in the sense that any Class 1 excitable system described by smooth ordinary differential equations (ODEs) can be transformed into this form by a continuous change of variables (Izhikevich, 1999). It takes only seven operations to simulate 1 ms of the model. It is highly recommended to choose this model when simulating large-scale networks of integrators (Izhikevich, 2004). The quadratic integrate-and-fire neuron model has more neurocomputational features than the LIF model, namely, it has spike latencies, activity-dependent threshold (which is $v_{threshold}$ only when $I=0$), and bistability of resting and tonic spiking modes (Izhikevich, 2004). This spiking model has a spike generation mechanism, i.e., a regenerative upstroke of the membrane potential, (unlike the LIF model) (Izhikevich, 2007).

3. The mathematical description

The model can be written in the following formula,

$$C\dot{v} = k * (v - v_{rest}) * (v - v_{threshold}) + I(t), \qquad if\ (v \geq v_{peak}), then\ v \leftarrow c$$

where C is the capacitance, the variable v represents the membrane potential of the neuron, v_{rest} and v_{thresh} are the resting and the instantaneous threshold potentials, respectively (when $I=0$), and $k>0$ is a parameter (Izhikevich, 2010). To avoid simulating ‘infinity’, the voltage trajectory is clipped at some sufficiently large value, v_{peak}, and reset it to a new value c.

4. Computer programs of the model

The implementation of the model is provided in the below Figures 1 - 2 (with inline comments).

```matlab
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  QIF neuron model %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% tstop (model's time in milliseconds) is the only required param
function V=QIF(tstop,varargin)
%  time step
if nargin>1   dt=varargin{1};
else     dt = 1; end%  default value
%  firing thresholdthe resting and the instantaneous threshold potentials
if nargin>2     V_reset=varargin{2};
else      V_reset=-80; end% default value
%  initial value of the voltage
if nargin>3    V_init=varargin{3};
else     V_init=0; end% default value
% input current
if nargin>4     I(1:tstop/dt)=varargin{4};
else      I(1:tstop/dt)=10; end% default value - unit: nA
% parameter capac describes the capacitance
if nargin>5    capac=varargin{5};
else     capac=1;end % default value
% parameter k
if nargin>6   k=varargin{6};
else     k = .02;end %  default value
% parameter c describes the after-spike reset value of V
if nargin>7     c=varargin{7};
else      c=-80; end% default value
% parameter d  describes the after-spike reset value of U
if nargin>8   Vpeak=varargin{8};
else     Vpeak=10; end% default value
if nargin>9  V_th=varargin{9};
else     V_th=-40; end% default value

V = V_init;
V_trace = []; % voltage trace for plotting

for t = 1:(tstop/dt)
    if (V>=Vpeak)                    % did it fire?
        V = c;                     % voltage reset
    end;
    %implementation of the voltage equation
    V = V + (dt*(k*(V-V_reset)*(V-V_th)+I(t) )/capac);
    if (V>=Vpeak) V=Vpeak;end;
    V_trace = [V_trace V];
end
V=V_trace;
end
```

Figure 1. Matlab program for simulating the model.

The following Matlab script can be used to run the model with the default parameters:

```matlab
V=QIF(100);
```

where the model time is 100 ms and the other parameters are specified in the function in Figure 1 above.

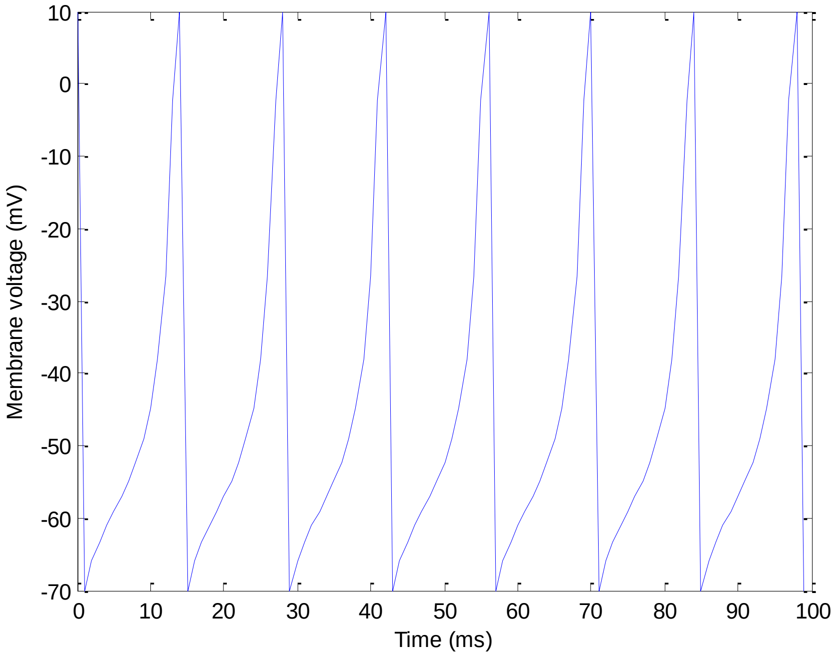

Figure 2. Voltage trace of the simulated model with a fixed input current. The plot can be reproduced using the MATLAB code 'plotmresults.m'

References

Ermentrout, B. (1996). Type I membranes, phase resetting curves, and synchrony. *Neural computation*, *8*(5), 979–1001.

Ermentrout, B., & Kopell, N. (1986). Parabolic Bursting in an Excitable System Coupled with a Slow Oscillation. *SIAM Journal on Applied Mathematics*, *46*(2), 233–253.

Izhikevich, E. M. (1999). Class 1 neural excitability, conventional synapses, weakly connected networks, and mathematical foundations of pulse-coupled models. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, *10*(3), 499–507.

Izhikevich, E. M. (2004). Which model to use for cortical spiking neurons? *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, *15*(5), 1063–70.

Izhikevich, E. M. (2007). Dynamical systems in neuroscience: The geometry of excitability and bursting. (T. J. Sejnowski & T. A. Poggio, Eds.)*Dynamical Systems*, *25*, 441.

Izhikevich, E. M. (2010). Hybrid spiking models. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, *368*(1930), 5061–70.

Shlizerman, E., & Holmes, P. (2012). Neural dynamics, bifurcations, and firing rates in a quadratic integrate-and-fire model with a recovery variable. I: Deterministic behavior. *Neural computation*, *24*(8), 2078–118.