**Matlab model code written in version R2020b**

clear all

% This code provides ascending and descending input to VIP cells and

% plots the response of all cell types in superficial (sup) and deep

% layers (inf), as shown in Fig. 3

% connectivity matrix

C = [0.006, -0.0144, -0.0029, -0.001, 0, -0.0026, -0.0034, 0;

0.304, -0.027, -0.00539, -0.00012, 0.03, -0.007, -0.0064, 0;

0.171, -0.01296, 0, -0.0016, 0, -0.0004, 0, 0;

0.26, -0.0035, -0.0053, -0.0011, 0, -0.0006, -0.01344, 0;

0.044, -0.00096, -0.0004, 0, 0.006, -0.01978, -0.004, 0;

0.104, -0.0089, -0.0002, 0, 0.132, -0.0773, -0.00896, 0;

0.055, 0, 0, 0, 0.04, -0.0041, 0, -0.0032;

0, -0.0013, -0.0058, 0, 0, 0, -0.0079, 0];

%g-value

g = [150];

%simulation duration in seconds

duration = 2;

%noise level

sigma = 0;

%synaptic time constants

tauE\_sup = 0.003;

tauPV\_sup = 0.007;

tauSST\_sup = 0.03;

tauVIP\_sup = 0.01;

tauE\_inf = 0.003;

tauPV\_inf = 0.007;

tauSST\_inf = 0.03;

tauVIP\_inf = 0.01;

% baseline external input

IextE\_sup\_baseline = 5;

IextPV\_sup\_baseline = 0;

IextSST\_sup\_baseline = 5;

IextVIP\_sup\_baseline = 0;

IextE\_deep\_baseline = 5;

IextPV\_deep\_baseline = 0;

IextSST\_deep\_baseline = 5;

IextVIP\_deep\_baseline = 0;

N\_scale\_vector\_up = 1:1:100;

N\_scale\_vector\_down = 100:-1:1;

columns = 1;

units = 8;

dt = 0.0001;

ds = 1;

tau\_vector = [tauE\_sup; tauPV\_sup; tauSST\_sup; tauVIP\_sup; tauE\_inf; tauPV\_inf; tauSST\_inf; tauVIP\_inf];

r(1:units,1) = 0.01;

c\_index = 0;

C\_scaled = C.\* g;

index = 0;

Rate\_vector\_up = zeros(units,numel(N\_scale\_vector\_up));

Rate\_vector\_down = zeros(units,numel(N\_scale\_vector\_down));

for n\_up = N\_scale\_vector\_up

% external input

IextE\_sup = IextE\_sup\_baseline ;

IextPV\_sup = IextPV\_sup\_baseline;

IextSST\_sup = IextSST\_sup\_baseline;

IextVIP\_sup = IextVIP\_sup\_baseline + n\_up;

IextE\_inf = IextE\_deep\_baseline;

IextPV\_inf = IextE\_deep\_baseline;

IextSST\_inf = IextE\_deep\_baseline;

IextVIP\_inf = IextE\_deep\_baseline + n\_up;

Iext\_vector = [IextE\_sup; IextPV\_sup; IextSST\_sup; IextVIP\_sup; IextE\_inf; IextPV\_inf; IextSST\_inf; IextVIP\_inf];

[Rate, Time] = microcircuit\_model(duration, columns, units, C\_scaled, tau\_vector, Iext\_vector, sigma, dt, ds, r);

index = index + 1;

Rate\_vector\_up(1:units,index) = Rate(1:units,10000);

r = Rate(:,size(Rate,2));

end

index = 0;

for n\_down = N\_scale\_vector\_down

% external input

IextE\_sup = IextE\_sup\_baseline;

IextPV\_sup = IextPV\_sup\_baseline;

IextSST\_sup = IextSST\_sup\_baseline;

IextVIP\_sup = IextVIP\_sup\_baseline + n\_down;

IextE\_inf = IextE\_deep\_baseline;

IextPV\_inf = IextE\_deep\_baseline;

IextSST\_inf = IextE\_deep\_baseline;

IextVIP\_inf = IextE\_deep\_baseline + n\_down;

Iext\_vector = [IextE\_sup; IextPV\_sup; IextSST\_sup; IextVIP\_sup; IextE\_inf; IextPV\_inf; IextSST\_inf; IextVIP\_inf];

[Rate, Time] = microcircuit\_model(duration, columns, units, C\_scaled, tau\_vector, Iext\_vector, sigma, dt, ds, r);

index = index + 1;

Rate\_vector\_down(1:units,index) = Rate(1:units,10000);

r = Rate(:,size(Rate,2));

end

figure;

colors = ['b', 'r', 'g', 'm', 'b', 'r', 'g', 'm'];

title\_plot = ["PYR-sup" "PV-sup" "SST-sup" "VIP-sup" "PYR-deep" "PV-deep" "SST-deep" "VIP-deep"];

plot\_index = 0;

for x = 1:8

plot\_index = plot\_index + 1;

subplot(2,4,plot\_index)

plot(N\_scale\_vector\_down, Rate\_vector\_down(x,:), 'k')

hold on

plot(N\_scale\_vector\_up, Rate\_vector\_up(x,:), colors(x))

title(title\_plot(x))

if or(plot\_index == 1, plot\_index==5)

ylabel('rate')

end

if plot\_index>=5

xlabel('Iext to VIP')

end

end