Appendix
Mathematical analysis of competing neural columns

Yihe Dong

Contents

This appendix provides a more systematic view of the most common parameter sets used for the different models in text, as well as selected *Mathematica, Matlab*, and *XPPAUT* codes used during the simulations and computations.

1 Parameter sets

Parameters in common for all models: $\tau_I = 5\ ms; \tau_N = 100\ ms; c_E = 310\ (VnC)^{-1}; c_I = 615\ (VnC)^{-1}; Io = 0\ nA; I_{thE} = 125\ Hz; I_{thI} = 177\ Hz; Is = .26\ A; a = 64/100; g_I = .087\ ms; g_E = .16\ ms; \tau_{Eref} = 2\ ms; \tau_{Iref} = 1\ ms.$

Ten dimensional model:$\tau_{rE} = \tau_{rI} = 0.1; J_{NEE} = 2.2\ nA; J_{GII} = 10\ nA; J_{NEI} = 1.7\ nA; J_{GIE} = 10\ nA; J_{AEE} = 0.65\ nA; J_{AEI} = 2.2\ nA; J_{AEE2} = 1.6\ nA; J_{AEI2} = 1\ nA; J_{NEE2} = 0.2\ nA; J_{NEI2} = 1\ nA$

Six dimensional model: (same as for the ten dimensional model; without τ_{rE}, τ_{rI})

Three dimensional model: $J_{NEE} = 6\ nA; J_{GII} = .05\ nA; J_{NEI} = 2\ nA; J_{GIE} = 3.5\ nA; J_{AEE} = 3.28\ nA; J_{AEI} = 4.5\ nA$

Two dimensional model:$J_{NEE} = 5.5\ nA; J_{GII} = .05\ nA; J_{NEI} = 3\ nA; J_{GIE} = 1.9\ nA;$

2 Center manifold calculation in 3D model

The following mathematica code follows the steps outlined in text for the center manifold calculation in the three dimensional model. This includes transformation into the standard form.

```
Clear[s1, s2, s3]

(*define parameters and input output functions*)
ti = 5; te = 100; ce \
= 310; ci = 615; Io = 0;
eIth = 125; iIth = 177; Is = 0.26; a = 64/100; gi = .087;
ge = .16; tre = 2; tri = 1; ta = 2;
Ie1 = Jnee*s1 + Jaee*s3 - Jgie*s2 + Io + Is;
Ii1 = Jaei*s3 + Jnei*s1 - Jgii*s2 + Io;
Fe[Ie1_] := (ce Ie1 - eIth)/(1 - Exp[-ge (ce Ie1 - eIth)] +
     tre/1000 (ce Ie1 - eIth));
Fi[Ii1_] := (ci Ii1 - iIth)/(1 - Exp[-gi (ci Ii1 - iIth)] +
     tri/1000 (ci Ii1 - iIth));
```

```
Jaee = 3.28; Jnei = 2; Jgii = 0.05; Jgie = 3.5; Jaei = 4.5; Jnee = 6;
S1 = -s1/te + a/1000*(1 - s1)*Fe[Ie1];
S2 = -s2/ti + Fi[Ii1]/1000;
S3 = -s3/ta + Fe[Ie1]/1000;
jac = {{D[S1, s1], D[S1, s2], D[S1, s3]}, {D[S2, s1], D[S2, s2],
    D[S2, s3]}, {D[S3, s1], D[S3, s2], D[S3, s3]}};

s1 = 0.18467123950038425; s2 = 0.2882515736499437; s3 = \
0.007078097221605036;

jac

{{0.306818, -0.186132, 0.174432}, {1.06848, -0.226712,
  2.40408}, {0.611493, -0.356704, -0.165717}}

ev = Eigensystem[jac]

{{0.0000167639 + 0.931091 I,
  0.0000167639 - 0.931091 I, -0.0856446}, {{0.103711 + 0.145167 I,
   0.936709 + 0. I, 0.0422471 + 0.298264 I}, {0.103711 - 0.145167 I,
   0.936709 + 0. I, 0.0422471 - 0.298264 I}, {0.481072,
   0.861344, -0.163267}}}

Tt = {{0.14516687954016247, 0.10371101443161514,
    0.48107155137713176}, {0, 0.9367091735935678,
    0.8613442485329903}, {0.2982644922224704,
    0.04224712244715756, -0.1632674124702721 3}};

TtI = Inverse[Tt];

Clear[s1, s2, s3]
DS2[x_, y_, z_] =
 1/2 (D[S2, {s1, 2}] (x^2) + D[S2, {s2, 2}] (y^2) +
      D[S2, {s3, 2}] (z^2)) + D[S2, s1, s2] x y + D[S2, s1, s3] x z +
   D[S2, s2, s3] y z +
   1/6 (D[S2, {s1, 3}] x^3 + D[S2, {s2, 3}] y^3 +
      D[S2, {s3, 3}] z^3) +
   1/2 (D[S2, s1, s1, s2] x^2 y + D[S2, s1, s2, s2] y^2 x +
      D[S2, s1, s1, s3] x^2 z + D[S2, s1, s3, s3] z^2 x +
      D[S2, s2, s2, s3] y^2 z + D[S2, s2, s3, s3] z^2 y) +
   D[S2, s1, s2, s3] x y z /. {s1 -> 0.18467123950038425,
   s2 -> 0.2882515736499437, s3 -> 0.007078097221605036}

0.0110209 x y - 0.991883 x z + 0.0247971 y z + 8.939 x y z +
 1/2 (-0.440837 x^2 - 0.000275523 y^2 - 2.23174 z^2) +
 1/2 (3.97289 x^2 y - 0.0993222 x y^2 - 357.56 x^2 z -
    0.223475 y^2 z - 804.51 x z^2 + 20.1127 y z^2) +
 1/6 (-158.915 x^3 + 0.00248305 y^3 - 1810.15 z^3)
DS3[x_, y_, z_] =
 1/2 (D[S3, {s1, 2}] (x^2) + D[S3, {s2, 2}] (y^2) +
      D[S3, {s3, 2}] (z^2)) + D[S3, s1, s2] x y + D[S3, s1, s3] x z +
```

```
   D[S3, s2, s3] y z +
   1/6 (D[S3, {s1, 3}] x^3 + D[S3, {s2, 3}] y^3 +
      D[S3, {s3, 3}] z^3) +
   1/2 (D[S3, s1, s1, s2] x^2 y + D[S3, s1, s2, s2] y^2 x +
      D[S3, s1, s1, s3] x^2 z + D[S3, s1, s3, s3] z^2 x +
      D[S3, s2, s2, s3] y^2 z + D[S3, s2, s3, s3] z^2 y) +
   D[S3, s1, s2, s3] x y z /. {s1 -> 0.18467123950038425,
   s2 -> 0.2882515736499437, s3 -> 0.007078097221605036}

-46.8416 x y + 43.8973 x z - 25.6068 y z - 1383.79 x y z +
 1/2 (80.3 x^2 + 27.3243 y^2 + 23.9972 z^2) +
 1/2 (-2531.33 x^2 y + 1476.61 x y^2 + 2372.22 x^2 z +
    807.213 y^2 z + 1296.81 x z^2 - 756.474 y z^2) +
 1/6 (4339.42 x^3 - 861.356 y^3 + 708.924 z^3)

(*transform coordinates*)Ty = Tt.{{y1}, {y2}, {y3}}

{{0.145167 y1 + 0.103711 y2 + 0.481072 y3}, {0.936709 y2 +
   0.861344 y3}, {0.298264 y1 + 0.0422471 y2 - 0.163267 y3}}

Fy = TtI.{DS1[Ty[[1]], Ty[[2]], Ty[[3]]],
    DS2[Ty[[1]], Ty[[2]], Ty[[3]]], DS3[Ty[[1]], Ty[[2]], Ty[[3]]]};

A = TtI.jac.Tt;

Dh[y1_, y2_] = {2 al y1 + b y2, b y1 + 2 c y2};

Fy2 = Fy /.
   y3 -> al y1^2 + b y1 y2 + c y2^2;(*Fy with substitution for y3*)

Ac = {{0, -0.9310905219583995}, {0.9310905219583995, 0}}.{{y1}, {y2}};

xDot[y1_, y2_] = Ac + {Fy2[[1]], Fy2[[2]]};

zDot[y1_, y2_] = -0.0856446 (al y1^2 + b y1 y2 + c y2^2) + Fy2[[3]];

Nh = Dh[y1, y2].xDot[y1, y2] - zDot[y1, y2];
Collect[Expand[Nh], {y1^2, y1 y2, y2^2}];

Fy3 = Fy2 /. {al -> 4.955446366369574, b -> -0.16866719757202356,
    c -> 4.577545140088854};

(*truncate at third order*)Collect[Expand[Fy3], {y1, y1 y2, y2}]

(*truncated*)
Fy4 = {{-35.29603590759219` y1 y2 + 24.07431242499003` y2^2 -
     117.78692052805945` y2^3 + y1^3 (29.039457310296473`) +
     y1^2 (12.991310369379159` + 73.94403528479307` y2) +
     y1 (440.93863474391867` y2^2)}, {+0.5930236605138126` y1 y2 -
     0.5121437455418462` y2^2 + 1.4315968297904842` y2^3 +
     y1^3 (-17.35722649842294`) +
```

```
     y1^2 (-0.4026357053264141` - 56.12498275265023` y2) +
     y1 (-22.51474421412378` y2^2)}};

(*center manifold approximation*)CM = Ac + {Fy4[[1]], Fy4[[2]]}

(*stability coefficient*)

alpha = 1/
     16 (D[Fy4[[1]], y1, y1, y1] + D[Fy4[[1]], y1, y2, y2] +
      D[Fy4[[2]], y1, y1, y2] + D[Fy4[[2]], y2, y2, y2]) +
   1/(16*.931091) (D[Fy4[[1]], y1,
        y2] (D[Fy4[[1]], y1, y1] + D[Fy4[[1]], y2, y2]) -
      D[Fy4[[2]], y1,
        y2] (D[Fy4[[2]], y1, y1] + D[Fy4[[2]], y2, y2]) -
      D[Fy4[[1]], y1, y1] D[Fy4[[2]], y1, y1] +
      D[Fy4[[1]], y2, y2] D[Fy4[[2]], y2, y2]) /. {y1 -> 0, y2 -> 0,
   y3 -> 0}

{-117.941}
```

3 Standard form calculation for 2D model

```
(*compute two dimensional stability coefficient*)

In[36]:= ti = 5; te = 100; ce = 310; ci = 615; Io = 0;
eIth = 125; iIth = 177; Is = 0.26; a = 64/100; gi = .087;
ge = .16; tre = 2; tri = 1; ta = 2;
Ie1 = Jnee*s1 - Jgie*s2 + Io + Is;
Ii1 = Jnei*s1 - Jgii*s2 + Io;
Fe[Ie1_] := (ce Ie1 - eIth)/(1 - Exp[-ge (ce Ie1 - eIth)] +
     tre/1000 (ce Ie1 - eIth));
Fi[Ii1_] := (ci Ii1 - iIth)/(1 - Exp[-gi (ci Ii1 - iIth)] +
     tri/1000 (ci Ii1 - iIth));
Jnei = 3; Jgii = 0.05; Jgie = 1.9; Jnee = 5.5;

In[44]:= S1 = -s1/te + a/1000*(1 - s1)*Fe[Ie1];
S2 = -s2/ti + Fi[Ii1]/1000;

jac = {{D[S1, s1], D[S1, s2]}, {D[S2, s1], D[S2, s2]}};

In[35]:= Clear[s1, s2]

In[47]:= FindRoot[{S1 == 0, S2 == 0}, {{s1, 0.185}, {s2, 0.288}}]

Out[47]= {s1 -> 0.143352, s2 -> 0.355912}

In[55]:= s1 = 0.1433515127433159`; s2 = 0.3559115637691499;

In[37]:= jac

Out[37]= {{-0.0117505, 0}, {0, -(1/5)}}
```

```
In[49]:= ev = Eigensystem[jac]

Out[49]= {{0.00226103 + 0.283072 I,
  0.00226103 - 0.283072 I}, {{0.140982 + 0.174579 I,
   0.974498 + 0. I}, {0.140982 - 0.174579 I, 0.974498 + 0. I}}}

In[50]:= Tt = {{0.17457947126975198` , 0.1409823181279271`}, {0,
    0.9744978164092774` }};

In[51]:= TtI = Inverse[Tt];

In[58]:= Clear[s1, s2]

In[53]:= (*second and third order derivatives for S1,here x,y \
represent small changes in the three coordinates*)
DS1[x_, y_] =
 1/2 (D[S1, {s1, 2}] (x^2) + D[S1, {s2, 2}] (y^2)) +
   D[S1, s1, s2] x y +
   1/6 (D[S1, {s1, 3}] x^3 + D[S1, {s2, 3}] y^3) +
   1/2 (D[S1, s1, s1, s2] x^2 y + D[S1, s1, s2, s2] y^2 x) /. {s1 ->
    0.1433515127433159`, s2 -> 0.3559115637691499`}

Out[53]= -11.2779 x y + 1/2 (32.3634 x^2 + 3.92978 y^2) +
 1/2 (-838.721 x^2 y + 294.327 x y^2) + 1/6 (2389.44 x^3 - 103.262 y^3)

In[59]:= (*second and third order derivatives for S1,here x,y \
represent small changes in the three coordinates*)
DS2[x_, y_] =
 1/2 (D[S2, {s1, 2}] (x^2) + D[S2, {s2, 2}] (y^2)) +
   D[S2, s1, s2] x y +
   1/6 (D[S2, {s1, 3}] x^3 + D[S2, {s2, 3}] y^3) +
   1/2 (D[S2, s1, s1, s2] x^2 y + D[S2, s1, s2, s2] y^2 x) /. {s1 ->
    0.1433515127433159`, s2 -> 0.3559115637691499`}

Out[59]= 0.0639996 x y + 1/2 (-3.83998 x^2 - 0.00106666 y^2) +
 1/2 (3.05811 x^2 y - 0.0509684 x y^2) +
 1/6 (-183.486 x^3 + 0.000849474 y^3)

In[56]:= TtI.jac.Tt

Out[56]= {{0.00226103, -0.283072}, {0.283072, 0.00226103}}

In[61]:= (*transform coordinates*)Ty = Tt.{{y1}, {y2}}

Out[61]= {{0.174579 y1 + 0.140982 y2}, {0.974498 y2}}

In[67]:= Fy = TtI.{DS1[Ty[[1]], Ty[[2]]], DS2[Ty[[1]], Ty[[2]]]};

In[68]:= Fy4 = Collect[Expand[ Fy], {y1, y1 y2, y2}]
```

```
Out[68]= {{12.2724 y1^3 + y1^2 (2.87348 - 41.6505 y2) -
   6.3583 y1 y2 + 3.68008 y2^2 + 48.4769 y1 y2^2 -
   18.457 y2^3}, {-0.166975 y1^3 + y1^2 (-0.0600488 - 0.357921 y2) -
   0.0858122 y1 y2 - 0.0306573 y2^2 - 0.255742 y1 y2^2 -
   0.060911 y2^3}}

In[73]:= alpha =
 1/16 (D[Fy4[[1]], y1, y1, y1] + D[Fy4[[1]], y1, y2, y2] +
      D[Fy4[[2]], y1, y1, y2] + D[Fy4[[2]], y2, y2, y2]) +
   1/(16*0.2830722164006525) (D[Fy4[[1]], y1,
        y2] (D[Fy4[[1]], y1, y1] + D[Fy4[[1]], y2, y2]) -
      D[Fy4[[2]], y1,
        y2] (D[Fy4[[2]], y1, y1] + D[Fy4[[2]], y2, y2]) -
      D[Fy4[[1]], y1, y1] D[Fy4[[2]], y1, y1] +
      D[Fy4[[1]], y2, y2] D[Fy4[[2]], y2, y2]) /. {y1 -> 0, y2 -> 0,
   y3 -> 0}

Out[73]= {-7.7571}
```

4 Sample Matlab code

The following code was part of the decision making simulation for the ten dimensional model.

```
%Two-column full 10D microcircuit
%output: time; gating variables & firing rates
function [T S] = TwoC10D3(Jnee,Jaee,Jnei,Jaei,Jgii,Jgie,Jnee2,Jaee2,Jnei2,Jaei2)
ge = .16;
dSS=ode45(@osciF,[0 1000],...
[0.000159361524115760;0.126528564327566;0.222519192119790;0.876854392886967;0.185493015887842;0.2225191

T=dSS.x;
S=dSS.y;

function dS=osciF(t,S)
ti = 5; te = 100;  ce = 310;  ci = 615; Io = 0 ;
eIth = 125; iIth = 177; Is = 0; Is2 = 0; a = 64/100; gi = .087;
ge = .16; tre = 2; tri = 1; ta= 2; tree=.1;trii=0.1;
dS=zeros(10,1);
%synaptic gating variables
Ie1=Jnee.*S(1)+Jaee.*S(3)-Jgie.*S(2)+Jnee2.*S(4)+Jaee2.*S(6)+Io+Is;
Ii1=Jaei.*S(3)+Jnei.*S(1)-Jgii.*S(2)+Jnei2.*S(4)+Jaei2.*S(6)+ Io;
Fe1=(ce.*(Ie1)-eIth)./(1-exp(-ge.*(ce.*(Ie1)-eIth))+tre./1000*(ce.*(Ie1)-eIth));
Fi1=((ci*(Ii1)-iIth)/(1-exp(-gi.*(ci.*(Ii1)-iIth))+tri./1000.*(ci.*(Ii1)-iIth)));

Ie2=Jnee.*S(4)+Jaee.*S(6)-Jgie.*S(5)+Jnee2.*S(1)+Jaee2.*S(3)+Io+Is2;
Ii2=Jnei.*S(4)+Jaei.*S(6)-Jgii.*S(5)+Jnei2.*S(1)+Jaei2.*S(3)+ Io;
Fe2=(ce.*(Ie2)-eIth)./(1-exp(-ge.*(ce.*(Ie2)-eIth))+tre./1000*(ce.*(Ie2)-eIth));
Fi2=((ci*(Ii2)-iIth)/(1-exp(-gi.*(ci.*(Ii2)-iIth))+tri./1000.*(ci.*(Ii2)-iIth)));

%dS(1),NMDA; dS(2),GABA; dS(3),AMPA; (same order for 4,5,6)
```

```
dS(1)= -S(1)./te+a./1000.*(1-S(1)).*S(7);
dS(2)= -S(2)./ti+S(9)./1000;
dS(3)= -S(3)/ta+S(8)./1000;

dS(4)= -S(4)./te+a./1000.*(1-S(4)).*S(8);
dS(5)= -S(5)./ti+S(10)./1000;
dS(6)= -S(6)/ta+S(8)./1000;

dS(7)=1/(tree)*(-S(7)+Fe1);
dS(8)=1/(tree)*(-S(8)+Fe2);
dS(9)=1/(trii)*(-S(9)+Fi1);
dS(10)=1/(trii)*(-S(10)+Fi2);
end
end
```

5 Sample .ode file for XPP and AUTO

XPP is a tool for numerically solving differential equations, boundary value problems, etc. The following sample was used to compute the nullclines for the three dimensional model.

```
#Reduced two-unit model with recurrent AMPA

#====Dynamical equations=======================================

sE1'=-(sE1/Tnmda)+(1-sE1)*gamma*nuE1(sE1,sI1,sE2)/1000
sI1'=-(sI1/Tgaba)+nuI1(sE1,sI1,sE2)/1000
sE2'=-(sE2/Tampa)+nuE1(sE1,sI1,sE2)/1000

#====Initial conditions========================================

init sE1=0.0798,sI1=0.1905,sE2=0.0067

#====Parameters================================================

#-----External stimulus-------
#par mu0=30.0,coh=0.0
par IstimE1=0.26,IstimI1=0

#----Synaptic constants ---

#NMDAR
par Tnmda=100,gamma=0.64

#GABAR
par Tgaba=5,Tampa=2

#----Input-output function parameters (for a pyramidal cell)--

par ce=310,ci=615,eIth=125,iIth=177,gi=0.087,ge=.16,tre=2,tri=1

#----Synaptic coupling constants--------------
```

```
par JNEE=6,JAEE=3.5,JNEI=2,JAEI=4.5,JGII=.05,JGIE=3.5

#----Constant effective external inputs + recurrent inhibition----

par I0=0

#-----Total synaptic input current to pyramdial cell----

xE1(sE1,sI1,sE2)=-JGIE*sI1 + I0 + IstimE1 + JAEE*sE2+JNEE*sE1
xI1(sE1,sI1,sE2)=-JGII*sI1 + I0 + IstimI1 + JAEI*sE2+JNEI*sE1

#-----Effective response function of E1 population------------
nuE1(sE1,sI1,sE2)=(ce*xE1(sE1,sI1,sE2)-eIth)/(1-exp(-ge*(ce*xE1(sE1,sI1,sE2)
-eIth))+tre/1000*(ce*xE1(sE1,sI1,sE2)-eIth))

#-----Effective response function of E2 population------------
nuI1(sE1,sI1,sE2)=(ci*xI1(sE1,sI1,sE2)-iIth)/(1-exp(-gi*(ci*xI1(sE1,sI1,sE2)
-iIth))+tri/1000*(ci*xI1(sE1,sI1,sE2)-iIth))

#======Setting of internal parameters for t===================

@ total=10000,dt=0.01
@ bound=10000,bell=off
@ background=white

#----Set the screen dimensions---------

@ xlo=-.5,xhi=1.2,ylo=-.5,yhi=1.5

#----Set the plotting variables--------

@ xp=SE1,yp=SI1

done
```