Biophysical Journal, Volume *120*

Supplemental information

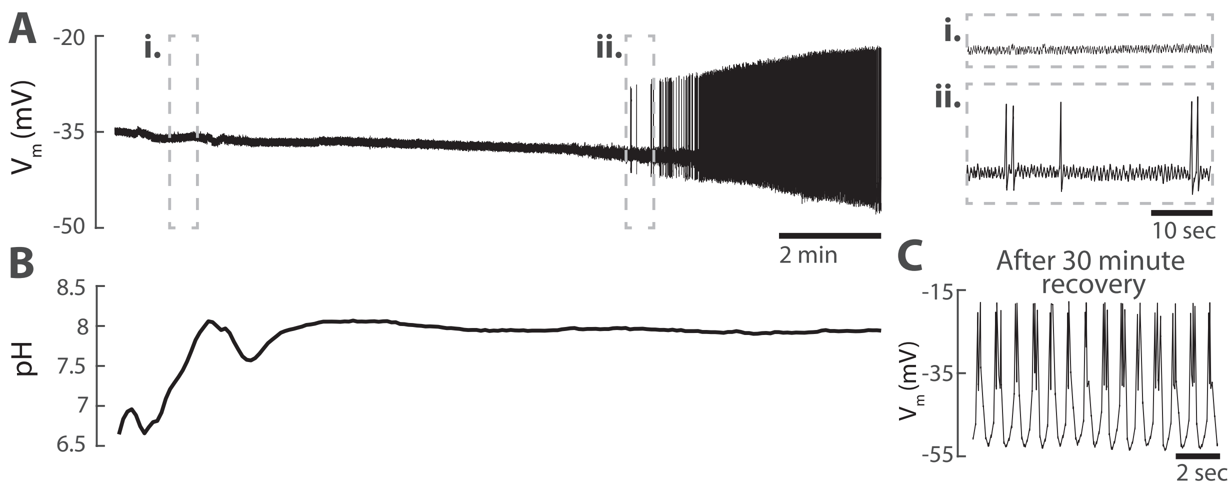
Neuronal oscillator robustness to multiple global perturbations

Jacob Ratliff, Alessio Franci, Eve Marder, and Timothy O'Leary

Neuronal oscillator robustness to multiple global perturbations

Jacob M. Ratliff, Alessio Franci, Eve Marder, Timothy O'Leary

Supplemental Materials



Supplemental Figure 1: Recovery of isolated oscillator in physiological pH. **A)** Intracellular recording from the PD neuron during recovery in physiological pH saline. Insets **i.** and **ii.** correspond to the same periods in left and right panels. **B)** pH recorded simultaneously with **A. C)** Intracellular recording from the PD neuron after 30 minutes in physiological saline.

Code for generating Figures 6 and 7

```
1. # MirroredFHN_interactive_tempH.jl
2.
3. using Plots, DifferentialEquations, Interact, Blink, LaTeXStrings
4.
5. # Run lines 1-85 to open an interactive window where you can
6. # play around with parameters and simulate the resulting behavior
7.
8. include("MirroredFHN_function.jl")
9.
10.  x0=zeros(3)
11.
12.  T=20000.
13.  tspan=(0,T)
14.
15.  ε=0.01
16.  εz=0.0001
17.
18.  ui = @manipulate for γ=-1.99:0.01:1.0, βf=-0.5:0.001:1.0, βs=-
   0.5:0.001:1.0,
19.      dIapp=-0.5:0.001:0.5
20.
```

```
    #βfmin=-2.:0.01:2., βfmax=-2.:0.01:2.,
    #βsmin=-2.:0.01:2., βsmax=-2.:0.01:2.,

    βfmin=βf-0.2;
    βfmax=βf+0.2;
    βsmin=βs-0.1;
    βsmax=βs+0.1;


    βfvec=range(βfmin,βfmax,length=10001);

    p1=plot(βfvec,lambdaTC.(γ,xTCm.(γ,βfvec)),label="TC variety",lw=2)
    p1=plot!(βfvec,lambdaTC.(γ,xTCp.(γ,βfvec)),label="TC variety",lw=2)
    p1=plot!(βfvec,lambdaHY.(βfvec,γ,xHY.(γ)),label="HY variety",lw=2)
    p1=plot!(βfvec,lambdaHYf.(βfvec,γ,xHYf.(γ)),label="fast HY variety", lw=2, legend=false,
        ylims=(βsmin,βsmax),grid=false)
        #xticks = βfmin:(βfmax+0.0001-βfmin)/20:βfmax+0.0001, xtickfont = font(5),
        #yticks = βsmin:(βsmax+0.0001-βsmin)/40:βsmax+0.0001,
        #ytickfont = font(5),dpi=150)
    p1=scatter!([βf],[βs],xlabel=L"\beta_f",ylabel=L"\beta_s")

    xvec=range(-1.5,1.5,length=250)

    p2=plot(xvec,-Ifast.(xvec,βf,βs,γ),lc=:black,label=L"I_{fast}")
    p2=plot!(xvec,-Islow.(xvec,βf,βs,γ),ylims=(-0.2,0.2),grid=false,lc=:gray,label=L"I_{slow}",
    legend=false,xlabel=L"V",)

    ITC=I_TC(βf,γ)
    if isnan(ITC)
        ITC=0.0
    end

    Iapp=ITC+dIapp

    p=(βf,γ,βs,Iapp,ε,εz)

    prob = ODEProblem(MirroredFHN_ODE!,x0,tspan,p)

    sol = solve(prob,BS3(),reltol=1e-3,abstol=1e-6);

    tt=minimum(findall(t->t>T/2,sol.t))
    p3=plot(sol.t[tt:length(sol.t)],sol[1,tt:length(sol.t)],
    legend=false,grid=false,ylims=(-1.33,1.0))

    xnullcline(x,y)= -x^3 + βf*x - 1/2*γ*x^2 - γ*βs*x - (y+βs)^2 - 1/2*γ*y^2
    ynullcline(x,y)= -y + x

    xvec=range(-1.1,1.0,length=250)
    yvec=range(-1.1,0.35,length=250)

    p4=contour(xvec,yvec,xnullcline,levels=[-ITC],lc=:red)
    p4=contour!(xvec,yvec,ynullcline,levels=[0],colorbar=false)
```

```julia

    l = @layout [
        [a{0.5w} b{0.5w}
        c{1.0w,0.75h}] d{0.5w}
    ]

    vbox(
    plot(p1,p2,p3,p4,layout=l,size=(1400,500))
    )

end

w = Window()
body!(w, ui)



## Static Drawing - In this part you can generate the static
# figures used to produce Figures 6 and 7
# uncomment the various parts of the code to define parameters
# as in figure 6 and being able to generate all the behaviors
# in that figure

include("MirroredFHN_function.jl")

γ=-0.1

βfmin=-0.1
βfmax=0.5
βsmin=0.1
βsmax=0.5

## perturbation paths
# Starting point (A)
# βf=0.3
# βs=0.15
# ITC=I_TC(βf,γ)
# if isnan(ITC)
#     ITC=0.0
# end
# Iapp=ITC-0.4

# Interm ph (B)
# βf=0.25
# βs=0.285
# ITC=I_TC(βf,γ)
# if isnan(ITC)
#     ITC=0.0
# end
# Iapp=ITC-0.25

# Interm ph 3 (C)
# βf=-0.05
# βs=0.35
# ITC=I_TC(βf,γ)
# if isnan(ITC)
#     ITC=0.0
```

```
# end
# Iapp=ITC-0.0


# Interm Temp 2 (D)
# βf=0.05
# βs=0.145
# ITC=I_TC(βf,γ)
# if isnan(ITC)
#     ITC=0.0
# end
# Iapp=ITC-0.3975

# Final temp (E)
βf=-0.033
βs=0.11
ITC=I_TC(βf,γ)
if isnan(ITC)
    ITC=0.0
end
Iapp=ITC-0.366

D=0.001
T=40000.
tspan=(0,T)

p1,p2,p3,p4,p5=plot_pchart(γ,βfmin,βfmax,βsmin,βsmax,βf,βs,Iapp,D,T,tspan)
#p1: parameter chart
#p2: IV curves
#p3: solution over time
#p4: phase plane
#p5: composite figure
plot(p3)
```

```
# MirroredFHN_function.jl

function MirroredFHN_ODE!(du,u,p,t)

    βf = p[1]
    γ = p[2]
    βs = p[3]
    Iapp = p[4]
    ε = p[5]
    εz = p[6]

    x = @view u[1]
    y = @view u[2]
    z = @view u[3]

    dx = @view du[1]
    dy = @view du[2]
    dz = @view du[3]

    @. dx = -x^3 + βf*x - 1/2*γ*x^2 - γ*βs*x - (y+βs)^2 - 1/2*γ*y^2 - z + Iapp
```

```
    @. dy = ε*(-y+x)
    @. dz = εz*(-z+x)

end

function MirroredFHN_σ(du,u,p,t)
    D=p[7]
    du[1]=0.0
    du[2]=D
    du[3]=0.0
end

## IV curves

Ifast(x,βf,βs,γ) = -x^3 + βf*x - 1/2*γ*x^2 - γ*βs*x
Islow(x,βf,βs,γ) = -x^3 + βf*x - 1/2*γ*x^2 - γ*βs*x - (x+βs)^2 - 1/2*γ*x^2
Iuslow(x,βf,βs,γ) = -x^3 + βf*x - 1/2*γ*x^2 - γ*βs*x - (x+βs)^2 - 1/2*γ*x^2 - x

## Transition varieties

function xTCm(gamma,beta)
    if gamma^4 + 48*beta>0
        f=( gamma^2 - (gamma^4 + 48*beta)^(1/2) )/( 12 )
    else
        f=NaN
    end
    return f
end

function xTCp(gamma,beta)
    if gamma^4 + 48*beta>0
        f=( gamma^2 + (gamma^4 + 48*beta)^(1/2) )/( 12 )
    else
        f=NaN
    end
    return f
end

lambdaTC(gamma,x)=-(x*(2+gamma))/2

xHY(gamma)=-(1+gamma)/3
lambdaHY(beta,gamma,x)=(beta-3*x^2-x*(2+2*gamma))/(2+gamma)

xHYf(gamma)=-gamma/6
lambdaHYf(beta,gamma,x)=(beta-gamma*x-3*x^2)/gamma

I_TC(beta,gamma)=-(-xTCm(gamma,beta)^3+beta*xTCm(gamma,beta)-
    (xTCm(gamma,beta)+lambdaTC(gamma,xTCm(gamma,beta)))^2-gamma*xTCm(gamma,beta)*(xTCm(gamma,beta)+lambdaTC(gamma,xTCm(gamma,beta))))
I_Hy(beta,gamma)=-(-xHY(gamma)^3+beta*xHY(gamma)-(xHY(gamma)+lambdaHY(beta,gamma,xHY(gamma)))^2)

## Plotting

```

```
function plot_pchart(γ,βfmin,βfmax,βsmin,βsmax,βf,βs,Iapp,D,T,tspan)

    βfvec=range(βfmin,βfmax,length=1000);

    p1=plot(βfvec,lambdaTC.(γ,xTCm.(γ,βfvec)),label="TC variety",lw=2)
    p1=plot!(βfvec,lambdaTC.(γ,xTCp.(γ,βfvec)),label="TC variety",lw=2)
    p1=plot!(βfvec,lambdaHY.(βfvec,γ,xHY.(γ)),label="HY variety",lw=2)
    p1=plot!(βfvec,lambdaHYf.(βfvec,γ,xHYf.(γ)),label="fast HY variety", lw=2, legend=false,
        ylims=(βsmin,βsmax),grid=false)
        #xticks = βfmin:(βfmax+0.0001-βfmin)/20:βfmax+0.0001, xtickfont = font(5),
        #yticks = βsmin:(βsmax+0.0001-βsmin)/40:βsmax+0.0001,
        #ytickfont = font(5),dpi=150)
    p1=scatter!([βf],[βs],xlabel=L"\beta_f",ylabel=L"\beta_s")

    xvec=range(-1.5,1.5,length=250)

    p2=plot(xvec,-Ifast.(xvec,βf,βs,γ),lc=:black,label=L"I_{fast}")
    p2=plot!(xvec,-Islow.(xvec,βf,βs,γ),ylims=(-0.2,0.2),grid=false,lc=:gray,label=L"I_{slow}",
    legend=false,xlabel=L"V",)

    p=(βf,γ,βs,Iapp,ε,εz,D)

    #prob = ODEProblem(MirroredFHN_ODE!,x0,tspan,p)
    #sol = solve(prob,BS3(),reltol=1e-3,abstol=1e-6);
    x0=zeros(3)
    prob = SDEProblem(MirroredFHN_ODE!,MirroredFHN_σ,x0,tspan,p)
    sol = solve(prob,saveat=10);

    #tt=minimum(findall(t->t>T/2,sol.t))
    p3=plot(sol.t[1:length(sol.t)],sol[1,1:length(sol.t)],
    legend=false,grid=false,ylims=(-1.1,1.0),xlims=(20000,40000))

    xnullcline(x,y)= -x^3 + βf*x - 1/2*γ*x^2 - γ*βs*x - (y+βs)^2 - 1/2*γ*y^2
    ynullcline(x,y)= -y + x

    xvec=range(-1.0,1.0,length=250)
    yvec=range(-1.1,0.35,length=250)

    p4=contour(xvec,yvec,xnullcline,levels=[-ITC+0*0.004],lc=:red)
    p4=contour!(xvec,yvec,ynullcline,levels=[0],colorbar=false)

    l = @layout [
        [a{0.5w} b{0.5w}
        c{1.0w,0.75h}] d{0.5w}
    ]

    p5=plot(p1,p2,p3,p4,layout=l,size=(1400,500))

    return p1,p2,p3,p4,p5

end
```