Note: For the c code in Figs 2-5 you will need gcc and gfortran or replace those commands in the makefile with your own compilers.
Python code for Fig 4 black circles and Fig 5d will need BRIAN2 toolbox installed

Figure 1: In c code directory
In biexpsum.c set TAU1 2.0 TAU2 0.3 P 4.0
cc biexpsum.c -lm -o biexpsum
biexpsum > biexptrain4.data

Figure 2: In c code directory
Important: In all cases for c code set E_SYN in avia.h (aviaprc only) or bvia.h to the desired value of -55 or -75 mV
Panel C
In bvia.h DELAY is 3 ms in all cases, first order resetting goes to f11.data, second order to f21.data
Commands for black traces
make mprcvia
mprcvia
Commands for black traces
make aprcvia
aprcvia
Panel A
In order to print the voltage and conductance traces for a single prc run
In bvia.h set DEBUG to 1
In mprcviadriver.c comment out line 92
If (!PRINT) {for(counter=0;counter<INCREMENT;counter++)
Uncomment line 93 and set the counter range as below
A1 counter range 56 to <57
A2 counter range 49 to <50
The commands are
make mprcvia
mprcvia > v.data
this will print the voltage waveform
then in pviaprcout.c comment out line 63
if (DEBUG) printf("%f %f/n"), *x,state[0]);
 and uncomment line 64 to print conductance waveform
again type:
make mprcvia
mprcvia> g.data
Panel B. aviaprc
In aprcviadriver.c comment out line 86
If (!PRINT) {for(counter=0;counter<INCREMENT;counter++)
Uncomment line 87 and set the counter range as below

B1 counter range 63 to <64
B2 counter range 27 to <28
Set PRINT to 1 in avia.h
Type
make aprcvia
aprcvia>v.data
then in aviaprcout.c comment out line 73
if (PRINT) printf("%f %f/n"), *x,state[0]);
 and uncomment line 72 to print conductance waveform

Figure 3B

Commands to type
make dprcvia
dprcvia
output goes to file pfpn.data

Figure 4

Top panel

green circles

command to to type
make freqpred
freqpred
output goes to the file prediction.data

red circles

in freqpred.c
comment out line 145 that begins if(oldpf<oldpn) and uncomment line 144

black circles

python3 spiketimes.py
prints frequencies to observation.txt

Bottom Panel

In line 41 of sviaprcdriver.c make sure the right file with the predicted frequencies is referenced
Commands to type
make sprcvia
sprcvia
the output goes to slope.data

Figure 5

Panel A Second order resetting was generated simultaneously with first order resetting from Fig 2 Panel C1

Panel B1 need to change delay from 3.0 to 5.2 in bvia.h to generate second order resetting from code for Fig 2 Panel C2

Panel B2 same code from B1 puts ts in ts.data

Panel C

Edit avia.h to set E_SYN -75 G_SYN3 36*0.0214844   DELAY 5.2 ms and FREERUN 1

In aprcviadriver.c comment out line 86
If (!PRINT) {for(counter=0;counter<INCREMENT;counter++)
Uncomment line 87 and set the counter range as below
B1 counter range 63 to <64

This simulates a fully self-connected oscillator approximating a synchronous network
Commands to type:
make aprcvia
aprcvia > v.data

Panel D

In singlerun.py set Es=-75 dv=4.9 randics True one_perturbed False
Command to type:
python3 singlerun.py

will generate Fig5d.eps
If you set randics False one_perturbed True
With random initial conditions you get synchrony instead
Spike times are written to spike_trains.txt