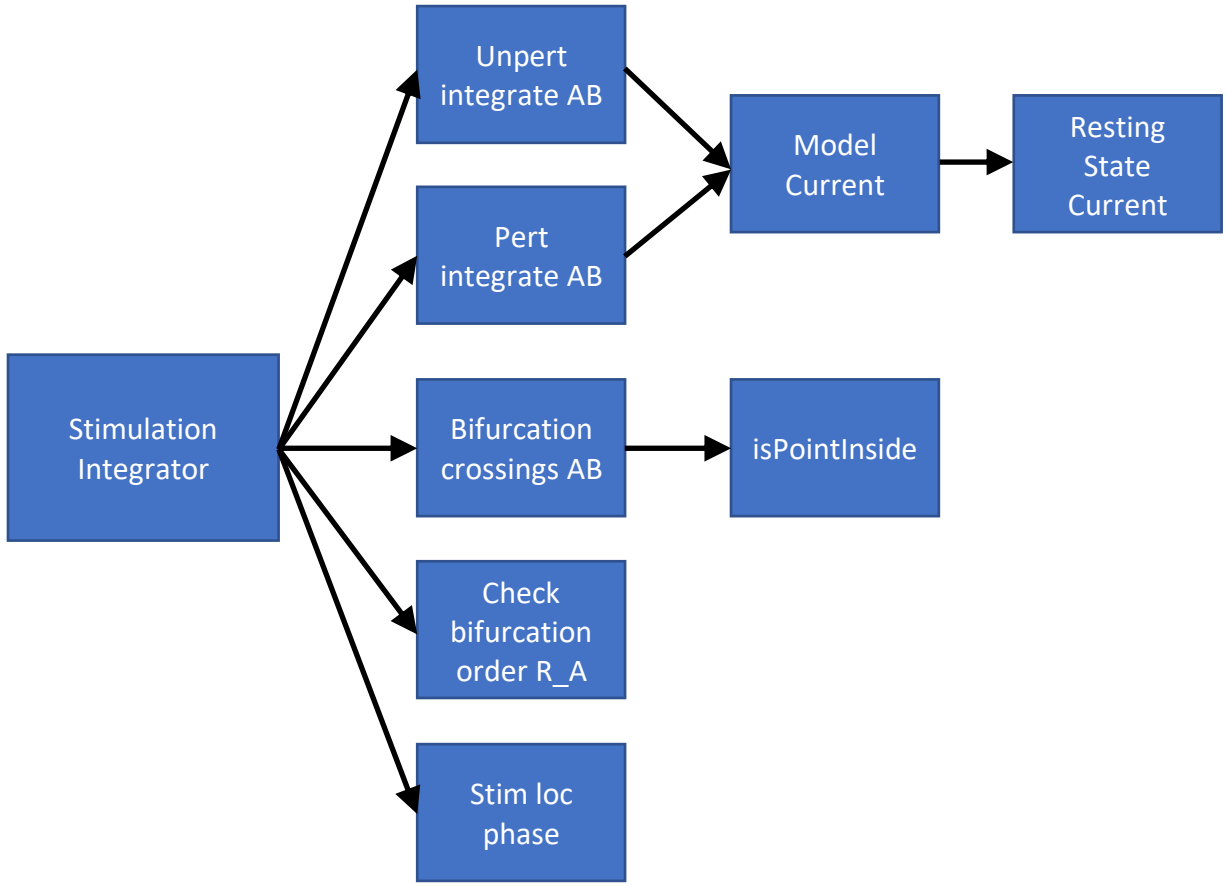
Model

The model is adapted from Saggio et al. 2017. Please refer to the link here: https://arxiv.org/abs/1605.09353 and familiarize yourself with the core concepts of this paper. See *ReadMe_Model.pdf* for a brief, high level summary of how we computationally employ the model.

If you would like to examine some simplified simulations of the onset/offset bifurcations in the model, and how different bursting paths generate these bifurcations, refer to this link: https://deepblue.lib.umich.edu/data/concern/data_sets/6q182k142.

Setup and Running the Script

The primary script is *Stimulation_Integrator.mlx*. This is where data is generated, and parameters are input. A flow chart with function calls represented as arrows is shown below. There is no need to open any other function if you wish to only generate data and look at examples.



Running Single Examples

If you would like to examine some pre-made examples of burster stimulation, I have included some simple instructions to do this:

1. Open *Stimulation_Integrator.mlx* and make sure the MATLAB path is set to the folder it resides in.

2. Make sure all functions in the flow chart above are in the folder, along with *Map.fig* and *Bifurcation_Curves_R_A.mat*.

3. When you open see that the code is broken into sections, which may include text and code (for those unfamiliar with live scripts). **The sections numbered (1)-(5) are the only sections that should be edited.**

4. Make sure the code in section (4) is uncommented and the code in sections (3) and (5) is commented. Section (5) is for generating large data sets (discussed below).

5. If you want to see the preset examples for an LCb burster (*c2b*), uncomment section (1) and comment section (2). Then, uncomment either subsection (a) or (b), depending on whether you want to see a stimulus which succeeds in terminating the burster or fails. The below image shows the setup for a *c2b* burster that will be unsuccessfully stimulated.

(1) LCb Burster with Successful and Unsuccessful Stimuli (*c2b*)

```
% DO NOT EDIT to make the below stimulus parameters applicable
class_numbers = 1; % see above
path_numbers = 1; % pick a number 1–5 for a unique path
```

(a) Unsucessful Stimulus Parameters:

```
    start_time = 0.5;
    amplitude = 3.0;
    pulse_width = 0.03;
```

(b) Sucessful Stimulus Parameters:

```
%     start_time = 0.5035;
%     amplitude = 3.0;
%     pulse_width = 0.03;
```

6. If you want to see the preset examples for an LCs burster (*c2s*), uncomment section (2) and comment section (1). Then, uncomment either subsection (a) or (b), depending on whether you want to see a stimulus which succeeds in terminating the burster or fails. The below image shows the setup for a *c2s* burster that will be successfully terminated:

(2) LCs Burster with Successful and Unsuccessful Stimuli (*c2s*)

```
% DO NOT EDIT to make the below stimulus parameters applicable
class_numbers = 5; % see above
path_numbers = 1; % pick a number 1-5 for a unique path
```

(a) Unsucessful Stimulus Parameters:

```
%     start_time = 0.49;
%     amplitude = 3.0;
%     pulse_width = 0.06;
```

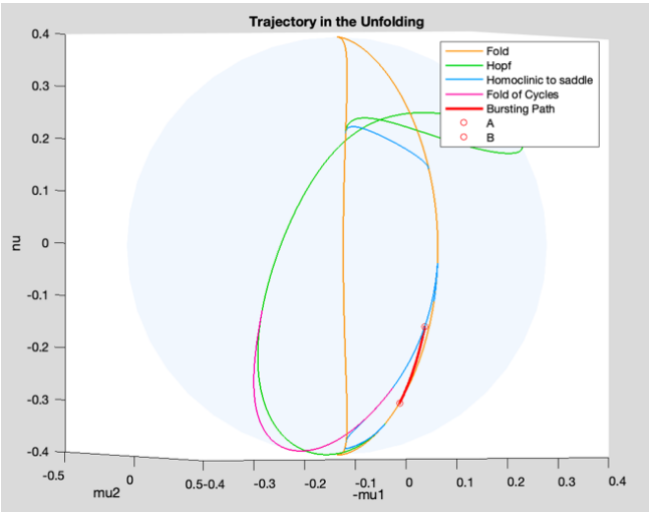
(b) Sucessful Stimulus Parameters:

```
    start_time = 0.5135;
    amplitude = 3.0;
    pulse_width = 0.06;
```

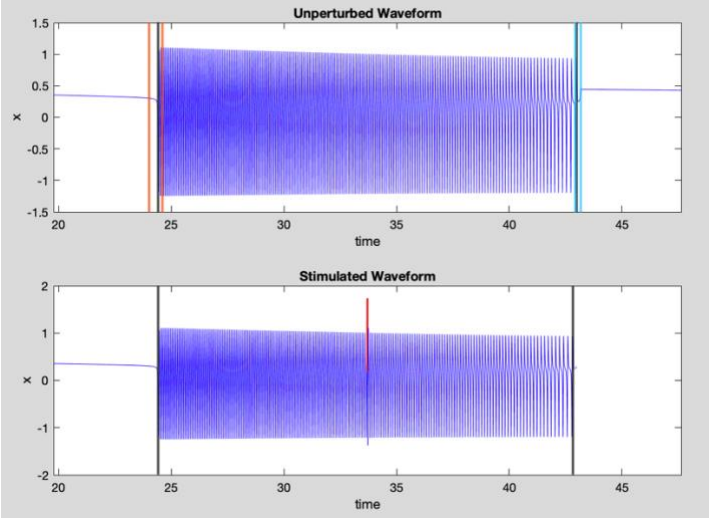
7. Once you have decided on the example you would like to see, run the whole script (press the green play button at the top).

8. After the script runs, three pop-up windows will appear with the following figures:

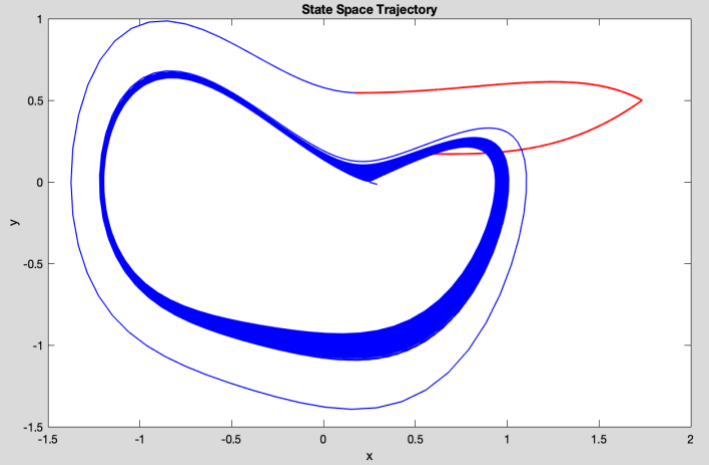
    a. The path the bursting path takes through parameter space with labeled bifurcation curves. An example is shown below:



    b. The voltage timeseries with and without stimulation. The original timeseries is on top with the beginning and end of burster showed by the black vertical lines. The estimated times at which the system bifurcates are shown by the colored lines: these colors match the color scheme in the figure above (i.e. saddle-nodes are orange). The bottom shows the timeseries with a stimulation (red if unsuccessful and green if successful). It also has black vertical lines showing where the bursting ends when stimulated. An example of an unsuccessful stimulus is shown below:



c. The trajectory of the system in state space with stimulation (red if unsuccessful and green if successful). You can see where on the limit cycle the stimulus is applied and the trajectory afterwards. The state space diagram corresponding to the voltage timeseries above is shown below:



If you want to experiment with changing the parameters:

1. Uncomment sections (3) and (4) and make sure sections (1), (2), and (5) are commented. In section (3) you can choose which class of burster you want to look at.

2. Set the variable “class_numbers” to an integer between 1 and 8. At the top of the script there is a numbering scheme for the classes.

3. Set the variable “path_numbers” to an integer between 1 and 5. I have created 5 different bursting paths for each class that you can choose from.

4. Set the variable “start_time” a real number between 0 and 1. This is when the stimulus will be applied. 0 corresponds to the beginning of the burster and 1 corresponds to the end.

5. Set the variable “amplitude” to any real number. This is the amplitude of the applied current stimulus. A negative number just flips the biphasic stimulus.

6. Set the variable “pulse_width” to a real number between 0 and 0.1. This is the ratio of the stimulus width to the inter-spike interval of the burster. We keep this relatively small so we do not change the dynamics of the system.

7. Run the script. The same figures shown above will be created.

Generating Data Sets

If you want to generate the data used to make the figures in the paper, comment sections (1)-(4) and uncomment section (5). Then simply run the script. Note that this run will take 1-2 days to complete. Because of the time it takes to generate this data, I have included the data sets used in a folder called “Data.” However, feel free to test that the code accurately reproduces these data sets. The rest of this section is intended to help those who wish to make their own unique data sets.

This script allows you to choose which of the classes you want to generate data and will create a .mat file for each class. For each class, you can choose to use any of the five bursting paths preset or create your own. The indexing system should be self-explanatory, just be sure to have values for A, B, c, and the manual start/end time if that setting is active. Choices for A and B are made easiest by using points on the respective onset (B) and offset (A) bifurcation curves, which are loaded from *Bifurcation_Curves_R_A.mat*. The code will also tell you if the path you input generates a valid burster and gives some information and plotting options to assist in troubleshooting.

For each class, you can also choose the base timestep and the stimulation timestep for integration. Having a separate simulation timestep allows one to apply stimuli of arbitrarily small widths while not having to use large timespan arrays. You can also choose the maximum time in the timespan, but it is best to just keep this number relatively large because the code will trim the timespan.

Furthermore, you can choose how to determine the “bursting window,” which in this context is the portion of the waveform where stimuli can be applied. We would like this window to capture most of the bursting activity. The reason there is not one single method to accomplish this that some bursters are generated with an active rest state. For simple bursters (c2s, c2b, c4b, c14b, c16b) bursting activity begins when the system departs from the resting state. For these bursters, it is natural to use peaks and valleys in z(t) (the slow variable) to define the bursting window,

since this variable is essentially measuring distance from the resting state. However, for some LCs bursters (c3s, c10s, c11s) these peaks and valleys can correspond with shifts to or from a second fixed point (active rest). So, to capture the bursting, we either manually define a start/end time or use the estimated location of an onset/offset bifurcation. Note that if you use the bifurcation setting for start and end times, your bursting window corresponds to when the burster is bistable. This region is dynamically interesting, but not as relevant for clinical application when chosen as the bursting window. Also note that there is some error in the estimation of bifurcation times, and they may not perfectly represent when the system bifurcates, but they do a pretty good job.

Stimulus parameters can also be selected and modified in the three arrays corresponding to start time, amplitude, and width. Start times are listed as percentages of the burster window. Amplitudes are scaled by the maximum of amplitude of the burster oscillation. Width values are percentages of the mean inter-spike interval measured in the burster window; this is the α as discussed in the paper. It is best to keep values in the width vector well below 1 so as to not introduce confounding dynamical effects.

There is also a setting to adjust the width of the stimulus as the inter-spike interval varies. This is done by measuring all the inter-spike intervals and creating a cubic spline. Then, instead of using the mean inter-spike interval to scale the width, we pull a value from the spline depending on the time of the stimulus. Additionally, you can choose whether to keep the amplitude constant or adjust the amplitude so that the injected charge remains constant.

Output Data

The script generates a .mat file which records stimulus parameters, bursting path information, and by what percentage the burster length is reduced. For the exact contents, see the script and associated comments in the code.

Additional Information

For questions on more detailed mechanisms of the code, please refer to descriptions in the function scripts or in the code comments. Any questions you have that cannot be answered by the information provided should be directed to mszuromi@umich.edu.