Modelling a Central Pattern Generator

M.I. Stefan, University of Edinburgh, UK

Collaborative International Undergraduate Workshop 2016

1 Learning Goals

After completing this session, you will be in a position to do the following:

Thinking	Consolidate what you have learnt in lectures this week Think through the building blocks of a Central Pattern Generator
Doing	Use the Copasi software package for biomedical modelling Simulate a Central Pattern Generator under varying conditions
Feeling	Appreciate how complexity emerges from simple building blocks Gain confidence in using computational models

2 Introduction

In this mostly self-paced tutorial, you will work through modelling Central Pattern Generators and simulating them on a computer.[1][2]

The tasks given to you here are deliberately hard, so don't worry if you find them tough, or if you don't get through all of the material in one session. Instructors and teaching assistants are here to help, so please ask them whenever you have a question or feel stuck. Raise your hand if you need to attract their attention. Of course, please also feel free to discuss with the students around you and to work in pairs or small groups.

Tasks that you should complete are typeset in **bold**. If you are following along well and want to look further and deeper into the matter, there are opportunities whenever you see something written in *italics*. Feel free to skip those parts if you don't want to go into that amount of detail.

[1] This module is loosely based on an earlier tutorial for modelling CPGs in the Genesis software by Crook and Cohen (1998). However, we use a different modelling software and will focus on different aspects of modelling CPGs.

[2] Lauren Kershberg provided constructive feedback on earlier drafts. Thank you!

All materials can be downloaded from the course website here:
`http://melaniestefan.net/CPG.html`

3 Introducing Copasi

Copasi (short for **Co**mplex **Pa**thway **Si**mulator) is a popular modelling and simulation software used in computational biology (Hoops et al., 2006). It was originally designed for modelling biochemical pathways, but can be used for all models where things interact and change over time in number or intensity. In our case, we will use it to model interactions between abstract representations of oscillating neurons (or networks of neurons).

- **Download and install Copasi.**
  Download files and installation instructions can be found here:
  `http://copasi.org/Download/`

4 A Central Pattern Generator with Two Coupled Oscillators

First of all, let's look at a Central Pattern Generator made up of two oscillating neurons (or groups of neurons). We are not concerned with the exact mechanism underlying oscillations in each neuron, we just know they oscillate at specific frequencies.

- **Get the model.**
  There are two different ways to get the model.

  1. From the course Dropbox: The file is in the folder called "Neural Circuit Modeling Workshop" and is called `coupled_oscillators.cps`. Make a local copy so you can edit and save the file as you want.
  2. Or go to `http://melaniestefan.net/CPG.html`, navigate to Course Materials, download the file `coupled_oscillators.cps` and save it somewhere on your computer. (Right-click + "save link as"). Your operating system might add the file extension `.txt` - delete or change that so the file extension is `.cps`

- **Open the model in Copasi.**
  Start Copasi. Go to `File -> Open` and navigate to the model you have just downloaded

- **Examine the structure of Copasi.**
  The way Copasi is organised, the user specifies parts of a model (for instance, interacting entities, or parameters that control their interactions). You then specify what you want your output to be (e.g. what variables to plot), and then you specify a task you want Copasi to do for you, e.g.

run a time course. In our case, we have pre-defined two "Species", a few "Global Quantities", one "Plot" and one "Time Course". We will explore these in the following.

- **Run a time course.**
  We have pre-defined two neurons (or clusters of neurons) called L1 and R1. Go to Time Course and click run. This will bring up a plot of the activities of L1 and R1 (normalised to be between -1 and +1). What behaviour can you observe? (You can click on the legend underneath the plot to show or hide individual curves).
  Wondering about the axis labels? Ignore them! These are a consequence of how Copasi was set up to model biochemical interactions. Since this is its historical origin, it will assume everything it deals with is measured in Molar and seconds, unless we tell it otherwise. In this tutorial, we are concerned with qualitative behaviours, not with exact numerical values, so we won't bother to adjust the axes.

- **Identify parameters.**
  If you look under "Global Quantities", you will find two quantities called omega1 and omega2. Click on them to see how they are defined. The "Simulation Type" variable is set to "fixed", which means that omega1 and omega2 are constants that won't change over the course of the simulation. What do these constants stand for? Change them and run more time courses to find out!
  If you are more interested in the details of the model, then also check out the definitions of phi1 and phi2 and how omega1 and omega2 are used in there. Note that for phi1 and phi2, the "Simulation Type" is not fixed, but something different. What is it and what does it do?

- **Set parameters**.
  You will find that the Global Quantity `a` is set at 0 at the moment. What happens if you change it? Try setting it to 10^{-5} (in Copasi, you can write either 0.00001 or 1e-5; the latter is less error-prone). What happens? What if you set it to 10^{-4}? Based on your findings, what aspect of the system is controlled by `a`?
  Again, if you want to know more about the maths behind this simulation, take a look at how `a` *is used in the definitions of h12 and h21. What are those?*

- **Modify coupling**.
  You should now have a functioning system of two in-phase coupled oscillators. What would you need to do to make the coupling out-of-phase, i.e. to have L1 active when R1 isn't (and vice versa)?

- **Plot phase space**.
  Phase space can be a confusing concept at first, but it is really useful for looking at dynamical systems. The idea is rather than plotting activities of the two neurons (or clusters) L1 and R1 on the y axis and time on the

x axis, you plot one activity against another, i.e. L1 on the x axis and R1 on the y axis. The question moves from "At a given time, what are R1 and L1 going to be?" to "At a given L1, what is R1 going to be?". Note that this latter question may have more than one answer.
To look at your model in phase space, create a new plot and select L1 for the x axis, and R1 for the y axis. In Copasi, you can do that by navigating to plot and adding a new one. You can then define a curve by clicking "New Curve", and select what you want to plot. L1 and R1 are in `Species -> Transient Concentrations`
Run the time course again and look at your phase space plot. What does it tell you? Although you haven't plotted time explicitly, can you guess in what direction time goes?

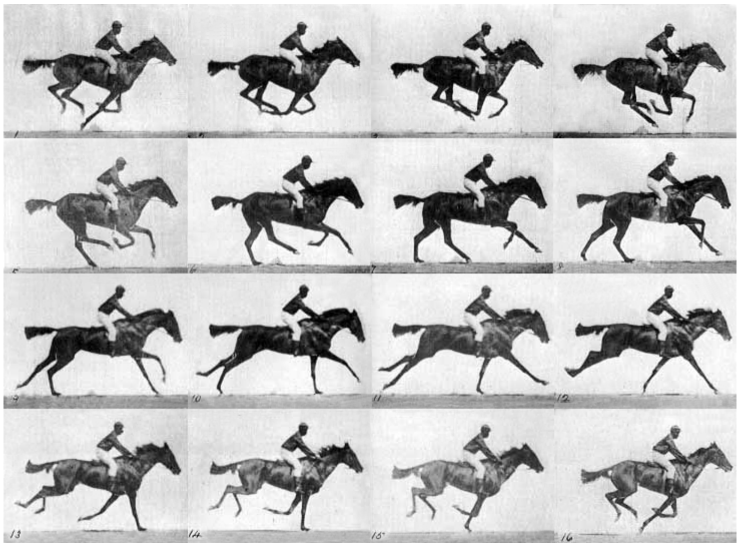
- **Introduce events**.
  One of the nice features of coupled oscillators is that they should find their way back into their coupled rhythm after the system has been disturbed (within reason). You can introduce a disturbance in your model by using the "Events" function in Copasi. An event has three components:

  1. The "trigger expression" tells the simulation what should trigger the event. This might be when the model time has reached a certain value, or when one of the concentrations or other parameter values has gone over an event threshold. In our case, choose a time that lies within your simulation time. You can do this by clicking on the little Copasi icon on the right of the box, and then select "model time". You can then type in an expression, for instance, type `== 100000` if you want to set the time of the event to 100 000 (note the double equal sign `==`).
  2. The "target" is the thing you want to change through your event. For instance, we will want change the value of either phi1 or phi2 (which control L1 and R1, respectively).
  3. The "expression" is the value you want to set for a target. In our case, it's enough to choose a simple number such as 2, but it might be a more complicated mathematical expression in other cases.

  What happens if you add an event to disturb L1 (or R1, or both) during the course of your simulation? How does this depend on coupling strength? What does it look like in your phase plot?

5 Modelling Animal Gait

Now, we want to model a Central Pattern Generator that can actually reproduce locomotion of a four-legged animal. Here is an image of a galloping horse (Muybridge, 1904).



- **Design a CPG to model animal gait**.
  In the following, we will be talking about a moving animal with four legs: two hind legs and two forelegs. We will model this as a system of four coupled oscillators, one for each limb. Sketch this system of four oscillators. Which of them have to be coupled? What type of coupling do you predict will result in what type of gait?

- ***Ninja mode: Create a model of four coupled oscillators***.
  You can do this by extending the model of two coupled oscillators that we used in the first part of the workshop. Do this if you found part one very easy. This takes a bit of thinking about parameter definitions - what stays the same, what changes? What gets connected to what and how is this expressed mathematically? You might also have to think carefully about the names you give to parts of the model in order to avoid confusion. Don't worry if you don't get all the way through it. A model we made is available for download at
  `http://melaniestefan.net/CPG/four_oscillators.cps`

- **Regular human mode: Download the model**.
  The model is called `four_oscillators.cps` - Again, you can find it online at
  `http://melaniestefan.net/CPG.html` or in the course dropbox.

- **Examine the model**.
  It is slightly bigger than the previous one, but a lot of the parts are the same or similar. Again, there are intrinsic oscillations for each of the four oscillators. There are now two coupling terms, `a_ba_fo` for coupling between the front and back legs and `a_le_ri` for coupling between the left and right side. Again, we have set those values to zero for now.
  If you want to examine exactly how the four oscillators are coupled, take a look at the definitions of the phi variables again.

- **Examine the effects of coupling**.
  It is now your turn to test different coupling settings and see what happens. You remember that you can have positive or negative and weak or strong coupling. Now you can also decide whether you want to couple front to back or left to right, or both. Play around with different options and carefully examine the plots you see. You might want to save plots generated under different conditions so you can compare them later. From the Copasi plot window, you can click on "Save image" to save a particular plot as an image. You can also use "Save data" to export the data you generated as a spreadsheet. This is especially useful if you want to document your work, want to perform further analysis on your data and/or make plots using different, more powerful plotting software.

- **Try a better way of plotting gaits**.
  The plot you have is great if you take time to examine and compare curves. But it is not extremely easy to interpret it at first sight as a plot of animal gait. This is because we are just plotting the activity of clusters of neurons, not their output (the actual action of limbs). This makes it difficult to distinguish, at first sight, a plot of a situation where right side and left side are positively coupled from a situation where they are not. Is there a better way to plot this? Sketch it out on paper and see if you can get Copasi to do it (the answer is probably yes - it might involve playing with plotting settings, and maybe even defining new Global Quantities.)
  Of course, if you are an experienced user of some other plotting software, you can also export your data from Copasi and visualise it elsewhere.

References

Crook, S. and Cohen, A. (1998). Central pattern generators. In *The Book of Genesis*, pages 131–147. Springer.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006). COPASI–a COmplex PAthway SImulator. *Bioinformatics*, 22(24):3067–3074.

Muybridge, E. (1904). Human and animal locomotion, plate 626, thoroughbred bay mare "annie g." galloping. Photograph. Public domain, edit by User:Waugsberg.