# Homework problem 6 – Whole Brain Models

**Deliverable:** One document (ideally PDF, MS Word is OK too) containing the answers to all questions and the figures specified in the task assignments below. You should not include all figures produced by the code. You do not have to include the modified code.

This last homework problem is meant to be a bit more hands-on and "real-life" than the previous exercises. The goal is to interact with publicly available code from a recent publication (Ponce-Alvarez and Deco, 2024).

Ponce-Alvarez, A., Deco, G. The Hopf whole-brain model and its linear approximation. *Sci Rep* **14**, 2615 (2024). https://doi.org/10.1038/s41598-024-53105-0

Take a brief look at the paper (i.e., read the abstract, admire the figures, do not worry about the math). You should have the paper at hand for comparison.

The code is available to download on github, which is where most people put their code these days. Github lets you keep track of any changes to your own code within a code repository. For now, we are just going to download the available Matlab code as a zip file. To do that, go here:

https://github.com/adrianponce/Linear-Hopf-model

and click on the green "<> Code" button. The last option is "Download ZIP". Select that and download the code.

Next, extract the zip file, open Matlab, and go to the unzipped code directory, which should be named "Linear-Hopf-Model-main".

In the following, we will only look at the model without delays between regions, including the following functions:

| | |
|---|---|
| StochSim_HopfNet.m | (simulates the model) |
| HopfModel_LNA.m | (linear approximation of the stochastic model) |
| Run_simulations_vs_LNA.m | (comparison between stochastic model and linear approximation) |

**Task 1:** Read the "README.md" file (You can do that in Matlab or with a text editor). What network size should you use to test the code? (1 pt)

The Hopf-model simulates whole-brain dynamics by coupling the local dynamics of N Hopf nodes, which are interconnected according to a given coupling matrix C representing anatomical connections.

**Task 2:** Load the Connectome20.mat file and visualize (plot) the coupling matrix using the Matlab command "imagesc". Include the figure in your document. (2 pts)

**Task 3:** Open the code for the stochastic simulation of the Hopf Model ("StochSim_HopfNet.m"). Set a break point on the last line of the code (l. 135). Run the Hopf model using the following parameters:
a = -0.5, g = 1, wo = 10, sigma = 0.3, T = 10, nTrials = 1

This is done like this:

```
StochSim_HopfNet(C, -0.5, 1, 10, 0.3, 10, 1)
```

The code should stop at your break point and all the variables used by the script should be in your workspace.

The variable X contains the real part of the activity of each node in the network, simulated over T = 10 s. In studies of whole-brain models, the brain signals (e.g. fMRI or MEG) are modelled by the real part of the state variables, i.e., x = Re(z), with

$$\frac{dz_j}{dt} = (a_j + \mathrm{i}\omega_j)z_j - |z_j|^2 z_j + g\sum_{k=1}^{N} C_{jk}(z_k - z_j) + \eta_j$$ (equation 6 in the paper).

Note that the whole network is first simulated for 30 s to let the system settle from a transient regime into the stationary regime (see code under “transient dynamics”).

Let’s plot the activity of each node as follows:

```
figure; plot(0:ds*dt:T,[1:20]+X)
```

Include the figure in the document. What is the purpose of adding [1:20] to X in the command above? (2 pts) [Hint: the figure should look somewhat like Fig. 1E in the paper, but for all 20 nodes]

**Q 1:** How would you go about finding suitable input parameters for **a**, **g**, **wo**, and **sigma** if you aren’t friends with the paper authors? (1 pt)

**Task 4:** Stop running the program (red stop button on top). Let’s replace the fixed values of “a” and “wo” by a vector of values drawn from a normal distribution with mean a = -0.2 and delta_a = 0.2, and mean wo = $2\pi$ and delta_wo = 0.1. To do this, first run

```
N = size(C,1);
a = -0.2 + 0.2*randn(N,1);
wo = 2*pi + 0.1*randn(N,1);
```

(Tip: If you don’t have C in your workspace you need to load Connectome20.mat again)

Then, run the `StochSim_HopfNet.m` script again, but with the “a” and “wo” vectors defined above and with sigma = 0.01. Include the command you used in your document.
Your code should again stop at the break point. Plot the node activity again as above and include the figure. Stop the script.
Next, run `StochSim_HopfNet.m` one more time with the same a and wo vectors as before and sigma = 0.01, but now with a coupling constant g = 0. Plot the node activity and include it in your document. (3 pts)

**Q 2:** What is the difference between the two activity plots with g = 1 and g = 0? What effect does the coupling have on the network? Look at Figure 2A in the paper. In our case, ||C|| = 1.1286 for the 20 node connectivity matrix. Where in the figure would the g = 1 and g = 0 simulation lie? (Hint: a_0 on the y-axis is the mean value of a). How does that explain the difference between the two activity plots? (4 pts)

**Task 5:** Remove the break point and stop running the program. Clear your workspace and close all figures. Open “run_simulations_vs_LNA.m”. Set the variable T in l. 34 to 300 and nTrials in l. 35 to 30. Run the script. This will take 1-3 min (if it takes significantly longer on your computer, you can decrease T further).

Include the resulting figures in your document. Compare your figures to Fig. 3A-E in the paper. Is the linear approximation for the 20 node network about as accurate as shown in the paper for the 1000 node network? (Note: you do not have to worry too much about what exactly is plotted here.) (2 pts)

**Task 6:** Fig. 6D shows the power spectral density (PSD) of a few example nodes. The PSD of a signal describes how the power of the signal is distributed across different frequencies.

Let’s find the height and location of the maxima for each node. We will use the analytically calculated PDA of the linear model to avoid the noise present in the stochastic simulation. The maxima and their locations can be obtained by running the following command:

```
[maxF, ind_max] = max(PSD_lna,[],1);
loc_max = x(ind_max);
```

Now make two new figures: 1.) plot the maxima against a, the bifurcation parameter of each node. 2.) Plot the location of the maxima against wo/(2*pi), the resonance frequency of each node. Include the figures in your document. What general trends can you observe in the two plots? (3 pts)

**Task 7:** In the “run_simulations_vs_LNA.m” script, set the global coupling parameter (l. 23) to 0. Run the script again. And repeat the last task (Task 6), with the results from our g = 0 simulation. Include the two resulting figures. What do you see now? Comparing the g = 0 figures to the figures from Task 6 with g = 3, what effect does the coupling have on the frequencies at which the nodes have the highest power? Does the coupling affect all nodes in the same way? (4 pts)

**Task 8:** Close all figures and clear your workspace. Set the coupling constant back to g = 3. Run the script one more time. We will now compare the functional connectivity produced by the model and the linear approximation against the original structural connectivity matrix.

The functional connectivity matrix of the stochastic simulation is computed from the simulated time course of the 20 nodes (see StochSim_HopfNet.m script). First, the average covariance between each pair of nodes is measured across the number of trials, which is then transformed into correlation coefficients in l. 133. In the linear approximation, the covariances (and thus the FC from the covariances) can be computed analytically.

Plot FC and FC_lna and the difference between FC and FC_lna using imagesc. Include the figures in your document. Now compute the correlation coefficients between FC, FC_lna, and SC. Since the matrices are symmetrical, we will use the lower triangle for comparison as follows:

```
corrcoef(tril(FC_lna,-1), tril(FC,-1))
```

Include the three values in your document (and label which is which). What can you conclude from the values? (5 pts)

**Task 9:** Final task. Do not clear your workspace. Let’s partition the model and see what effects this has on the functional connectivity values. Since the linear approximation works pretty well, we will use the approximation for this task. For our partition we will make the first ten nodes independent of the other ten nodes (Note: the 20 node connectivity matrix is simply a down-sampled version of the 1000 node matrix using every 50th value and unfortunately we do not know which brain region is which node).

To set the connections between the first 10 nodes and the rest to 0, run the following lines.

```
N = 20;
C_part = zeros(N);
C_part(1:10,1:10) = C(1:10,1:10);
C_part(11:20, 11:20) = C(11:20,11:20);
```

Plot C_part and include the figure in your document.
Now run the linearized Hopfmodel as follows:

```
FC_part = HopfModel_LNA(C_part,a,g,wo,sigma);
```

Note: all the variables for a, g, wo, sigma should still be the same as for the previous task.

Visualize FC_part with imagesc and include the figure in your document.

We would like to know what effect the partition had on the rest of the “brain”. To that end, we will compute the correlation between the partitioned and unpartitioned matrices, but only for the correlations within the intact clusters.

```
corrcoef(tril(FC_lna(1:10,1:10),-1), tril(FC_part(1:10,1:10),-1))
corrcoef(tril(FC_lna(11:20,11:20),-1), tril(FC_part(11:20,11:20),-1))
```

Which correlation is higher? Does that make sense given the original connectivity matrix?
Now correlate the FC_part with C_part. Is this correlation higher or lower than the correlation between FC_lna and C? What might that mean? (6 pts)

**Q 3:** Last question. If, instead, we partitioned the connections between every second element, would that likely have a larger or smaller impact on the simulated functional connectivity matrix based on the original connectivity matrix? Take a look at the connectivity matrix and explain. (< 20 words) (2 pts)

**(Total points: 35)**