Analysis of the Kuramoto Model with Network Couplings

Jonas Lauri

February 28, 2023

Contents

1 Introduction.

In this project we will consider a modified Kuramoto model where the couplings (between oscillators) are determined by a network, $\mathbf{A}$. The connections between the nodes are initialized randomly but will be optimized. Connections are unweighted,

$$A_{ij} = \begin{cases} 1, & \text{when a connection exists,} \\ 0, & \text{else.} \end{cases} \tag{1}$$

We will also have a phase shift, α, which in the beginning will be held constant but will later also be part of the analysis. The natural frequencies will be

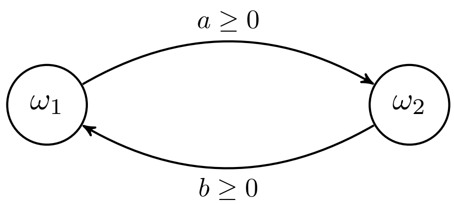


Figure 1: A graphical representation of the simple model.

initialized in an equidistant manner. The model will be,

$$\dot{\phi}_i = \omega_i + \frac{\kappa}{C}\sum_{j=1}^{N} A_{ij}\left[\sin\left(\phi_j - \phi_i + \alpha\right) - \sin(\alpha)\right], \tag{2}$$

where $\sum_{i,j} A_{ij} = C$ where C is some constant (and N is the number of nodes).

To evaluate how good the system is we will look at the r parameter. It is calculated like

$$r = \frac{1}{N}\left|\sum_{j=1}^{N} e^{i\phi_i}\right| \tag{3}$$

which is basically the vector sum of all the different oscillators. It expresses how in sync the different oscillators are at a given time. For example, if they are all clumped together we would get a $r \approx 1$ and if they are uniformly spread out we would get a $r \approx 0$.

2 Analytical analysis of the $N = 2$ case.

First consider a simple model with only two oscillators. From figure 1 we can write down the $\mathbf{A}$ of this model.

$$\mathbf{A} = \begin{pmatrix} 0 & a \\ b & 0 \end{pmatrix}, \quad a, b \geq 0. \tag{4}$$

From $\mathbf{A}$ we can get the equations for the $\dot{\phi}$.

$$\dot{\phi}_1 = \omega_1 + \frac{\kappa}{C}\, a\sin(\phi_2 - \phi_1 + \alpha) \tag{5}$$

$$\dot{\phi}_2 = \omega_2 + \frac{\kappa}{C}\, b\sin(\phi_1 - \phi_2 + \alpha) \tag{6}$$

Finally, look for fix points.

$$\dot{\phi}_1 = \dot{\phi}_2 \tag{7}$$

$$\omega_1 + \frac{\kappa}{C}\, a\sin(\phi_2 - \phi_1 + \alpha) = \omega_2 + \frac{\kappa}{C}\, b\sin(\phi_1 - \phi_2 + \alpha). \tag{8}$$

2.1 The case: $\underline{\omega_1 = 1,\, \omega_2 = 0,\, a = b}$

First let us use equation 8 with the simplified case above. This gives us

$$1 + \frac{\kappa}{C}\, a \sin(\phi_2 - \phi_1 + \alpha) = \frac{\kappa}{C}\, a \sin(\phi_1 - \phi_2 + \alpha). \tag{9}$$

Remember that the sine function is an odd function, $f(x) = -f(-x)$.

$$1 - \frac{\kappa}{C}\, a \sin(\phi_1 - \phi_2 - \alpha) = \frac{\kappa}{C}\, a \sin(\phi_1 - \phi_2 + \alpha). \tag{10}$$

Rewrite as

$$1 = \frac{\kappa}{C}\, a \sin(\phi_1 - \phi_2 - \alpha) + \frac{\kappa}{C}\, a \sin(\phi_1 - \phi_2 + \alpha). \tag{11}$$

Factor $\frac{\kappa a}{C}$

$$\frac{C}{\kappa a} = \sin(\phi_1 - \phi_2 - \alpha) + \sin(\phi_1 - \phi_2 + \alpha). \tag{12}$$

Sum formula for sine gives us

$$\begin{aligned} \sin(A + B) + \sin(A - B) &= \sin(A)\cos(B) + \cos(A)\sin(B) \\ &\quad + [\sin(A)\cos(B) - \cos(A)\sin(B)] \\ &= 2\sin(A)\cos(B), \end{aligned}$$

with $A = \phi_1 - \phi_2$ and $B = \alpha$ we get

$$\frac{C}{\kappa a} = 2\sin(\phi_1 - \phi_2)\cos(\alpha). \tag{13}$$

Now, what we want to do is solve the following minimization problem

$$\min_{a,b,\alpha} |\psi|, \quad \psi = \phi_1 - \phi_2. \tag{14}$$

Insert ψ into equation 13 to obtain

$$\frac{C}{\kappa a} = 2\sin(\psi)\cos(\alpha) \tag{15}$$

which gives us

$$\psi = \sin^{-1}\left(\frac{C}{2\,\kappa\, a \cos(\alpha)}\right). \tag{16}$$

All variables in the expression for ψ are given and we are therefore done with this simple case.

2.2 The case: $\underline{\omega_1 = 1,\, \omega_2 = 0,\, a \neq b}$

Again, let us use equation 8 with the case above. This gives us

$$1 + \frac{\kappa}{C}\, a \sin(\phi_2 - \phi_1 + \alpha) = \frac{\kappa}{C}\, b \sin(\phi_1 - \phi_2 + \alpha). \tag{17}$$

Continue as above to get

$$1 = \frac{\kappa}{C}\, a \sin(\phi_1 - \phi_2 - \alpha) + \frac{\kappa}{C}\, b \sin(\phi_1 - \phi_2 + \alpha). \tag{18}$$

Factor $\frac{\kappa}{C}$

$$\frac{C}{\kappa} = a \sin(\phi_1 - \phi_2 - \alpha) + b \sin(\phi_1 - \phi_2 + \alpha). \tag{19}$$

Insert ψ

$$\frac{C}{\kappa} = a \sin(\psi - \alpha) + b \sin(\psi + \alpha). \tag{20}$$

From here it is possible to find solutions, however, the expressions for the solutions are very long. In the end we will solve the minimization problem 14 numerically.

3 Numerical analysis of $N = 3, 4$ cases.

The optimal network structure has been examined in earlier studies. From the article, *Optimizing synchrony with a minimal coupling strength of coupled phase oscillators on complex networks based on desynchronous clustering*, we get some rules as to how the connections in a optimized network should be distributed [1]. These are the following:

(i) The deviations of nodes' frequencies from the mean value are linear with the nodes' degrees.

(ii) The oscillators form a bipartite network divided according to the frequencies of the oscillators.

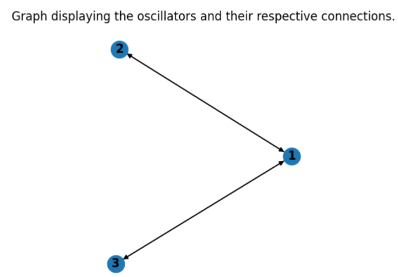
(iii) Oscillators are only connected to those with sufficiently large frequency differences.

The reason as to why we will only look at networks for $N = 3, 4$ is the rapid increase in the number of possible networks as we increase N. The number of possible networks are: $\frac{N!}{2}$ + trees. $\frac{N!}{2}$ comes from the possible permutations of N which is $N!$ The $\frac{1}{2}$ comes from the mirror symmetry. And as if this factorial increase was not enough, we also have to factor in the trees. It is easy to understand why this problem is as complex as it is.

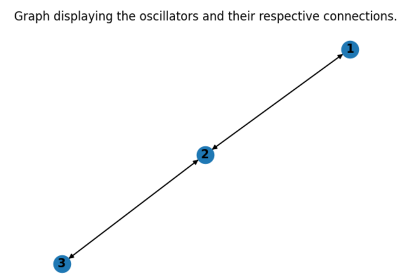
The distribution of inner frequencies, ω_i, are linearly spaced between zero and one. This applies to both the three networks and the four networks.

3.1 The case: $N = 3$.

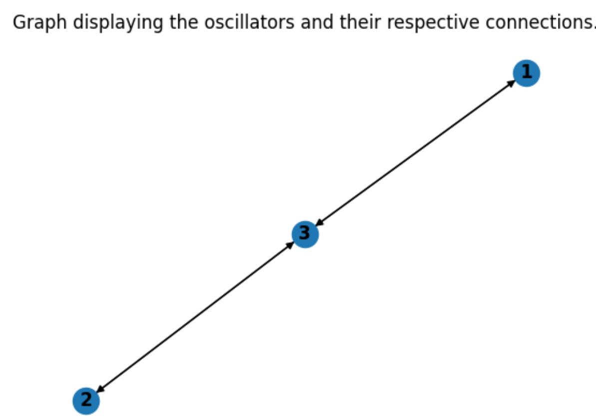
Below comes a couple simulations for the case where $N = 3$, $C = 2, \kappa = C$. For additional parameters please take a look at the file: `three_networks.py` The value of κ here is quite high. Therefore any connection will be sufficient to synchronize the oscillators. The resulting r-values can be seen in table 1.



(a) Node 1 center.



(b) Node 2 center.



(c) Node 3 center.

Figure 2: Graphical representation of the different networks with three nodes.

Network	Node 1 center	Node 2 center	Node 3 center
$\alpha = 0$	0.955	0.905	0.960
$\alpha = 0.1$	0.957	0.904	0.961
$\alpha = 0.3$	0.959	0.896	0.956
$\alpha = 0.5$	0.956	0.880	0.943
$\alpha = 1.0$	0.901	0.801	0.556
$\alpha = -0.5$	0.935	0.883	0.961
$\alpha = -1.0$	0.541	0.803	0.926

Table 1: Listed are the r-values for some different values of the phase shift, α. We see that a large phase shift is worse in general, but especially that the phase shifts affects the networks differently. The networks can be seen in figure 2.

3.2 The case: $N = 4$.

The case $N = 4$ is quite similar to the previous case. We use $N = 4, C = 3, \kappa = C$ where additional parameters are in the file: `four_networks.py`. The resulting r-values can be seen in table 2.

3.3 Optimal network structure.

Let us quickly take a look at if the above criteria for optimal network structure are being fulfilled for these simple networks.

First, let us take a look at (ii). Because of the small size of these networks it is hard to say anything for certain. Therefore, let us skip that for now.

For (i), let us take a look at the results in table 1. In table 1, we see that networks Node 1 center and Node 3 center outperform Node 2 center. The average inner frequency is a half and therefore Node 1 and Node 3 are the ones with the highest deviation from the mean. We see that the network where Node 1 and Node 3 has two connections are the best ones in terms of r-value.

For (iii) we can take a look a table 2. Left X is the network where (iii) is as realized as possible and Left X is also the best performing network. it seems to confirm the rule (iii).

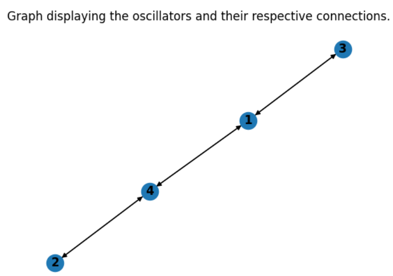
Finally, the results for alpha are the following. Positive alpha is better for networks where the nodes with large degree are the ones with small inner frequency. Negative alpha is better for networks where the nodes with large degree are the ones with large inner frequency.

4 Analysis of larger networks.

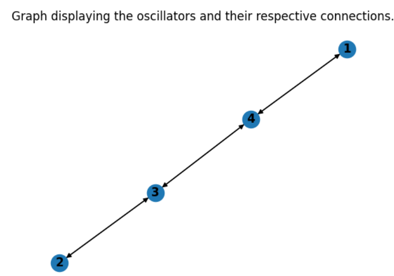
For the larger networks it is not feasible to list all possible networks as we have done with these smaller networks. Therefore the analysis will be more focused on how we can find the best networks. In this report we will focus on an

α	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1.0$
Star, root 1.	0.959	0.958	0.919
Star, root 2.	0.922	0.895	0.807
Star, root 3.	0.925	0.909	0.781
Star, root 4.	0.957	0.936	0.535
Right Snake, 1-2-3-4/4-3-2-1.	0.787	0.750	0.693
Up Snake, 4-1-2-3/3-2-1-4.	0.966	0.955	0.859
Left Snake, 3-4-1-2/2-1-4-3.	0.892	0.861	0.734
Down Snake, 2-3-4-1/1-4-3-2.	0.971	0.949	0.517
Right X, 1-3-2-4/4-2-3-1.	0.876	0.848	0.716
Down X, 1-3-4-2/2-4-3-1.	0.943	0.906	0.691
Left X, 2-4-1-3/3-1-4-2.	0.984	0.984	0.890
Up X, 3-1-2-4/4-2-1-3.	0.943	0.943	0.897
N, 4-1-3-2/2-3-1-4.	0.960	0.965	0.935
Z, 1-2-4-3/3-4-2-1.	0.843	0.762	0.649
Reverse N, 1-4-2-3/3-2-4-1.	0.959	0.921	0.519
Reverse Z, 2-1-3-4/4-3-1-2.	0.841	0.842	0.817

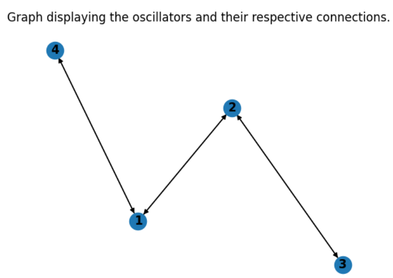
Table 2: Listed are the r-values for some different networks with a given phase shift, α. We see that a large phase shift is worse in general, but especially that the phase shifts affects the networks differently. The best six networks when $\alpha = 0$ can be seen in figure 3.



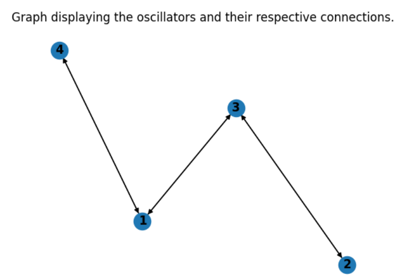
(a) Left X, 2-4-1-3/3-1-4-2.



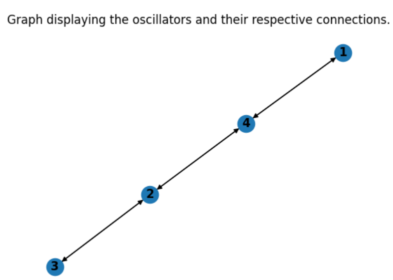
(b) Down Snake, 2-3-4-1/1-4-3-2.



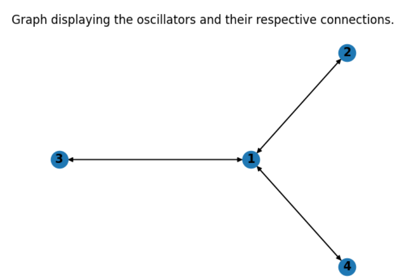
(c) Up Snake, 4-1-2-3/3-2-1-4.



(d) N, 4-1-3-2/2-3-1-4.



(e) Reverse N, 1-4-2-3/3-2-4-1.



(f) Star, root 1.

Figure 3: Graphical representation of some examples of the different networks with four nodes. They are listed in descending order with respect to their r-value when $\alpha = 0$.

optimization algorithm inspired by stochastic optimization methods, especially methods with inspiration from biology.

4.1 Optimization algorithm.

The general idea as to how we will optimize the connection matrix, $\mathbf{A}$, can be seen in algorithm 1. For general questions about the code we refer to `optimization.py`, but the updates of the $\mathbf{A}$-matrix are of great importance so they will be explained here. There are methods of adding/removing connections which cause the best results, however, here we will just add randomly. The number of connections being shuffled are called *updatesPerIteration* and is equal to 20% of the total number of connections.

Algorithm 1: A overview of the optimization algorithm.

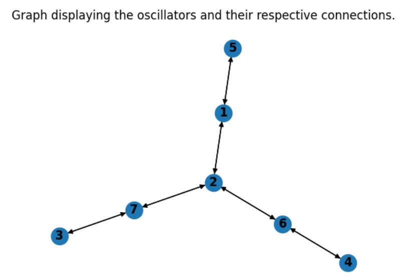
```
Initialization of A.
for timesteps do
    if iterationNumber > maxIterationsWithoutImprovement then
        A = bestA;
        iterationNumber = 0;
    end
    Evaluation of the model;
    Calculate r-value;
    if r > bestR then
        bestR = r;
        bestA = A;
        iterationNumber = 0;
    end
    Update the A-matrix;
end
```

4.2 Sample results from bigger networks.

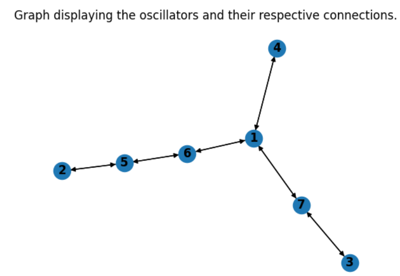
Let us now take a look at some results from these bigger networks. In figure 4 there four example networks from optimization. Additionally, there is a parameter, `is_directed`, which allows each connection to only go one way. In figure 4 we have also included two example networks where this is allowed. The results seem to follow the rules (i) - (iii).

4.3 Future developments.

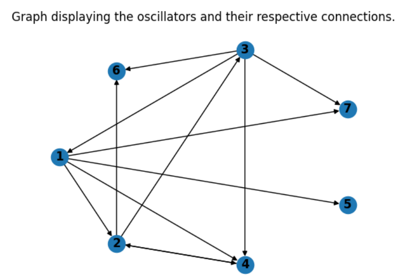
The biggest problem with the optimization in the current version is that it easily gets stuck. This problem is most easily fixed by choosing which connection to remove and add in a smarter way. A way in which this could be done is by choosing the eigenvector with the largest eigenvalue.



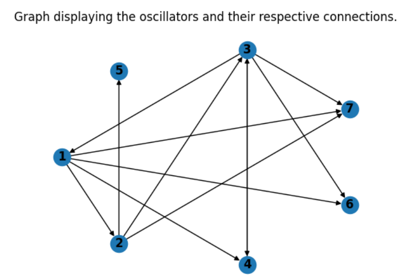
(a) First example of a undirected network with seven nodes, $r = 0.9869$.



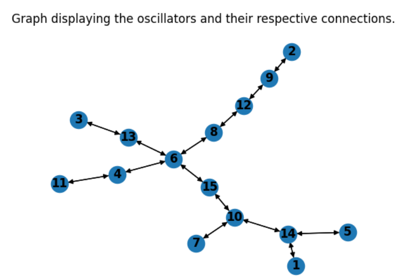
(b) Second example of a undirected network with seven nodes, $r = 0.9898$.



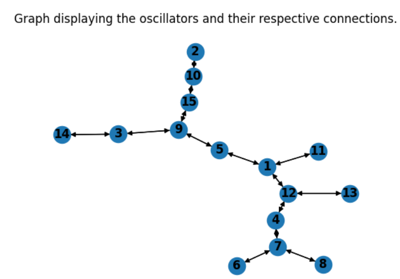
(c) First example of a directed network with seven nodes, $r = 0.9874$.



(d) Second example of a directed network with seven nodes, $r = 0.9942$.



(e) First example of a directed network with 15 nodes, $r = 0.9297$.



(f) Second example of a directed network with 15 nodes, $r = 0.9437$.

Figure 4: The resulting networks from 100 iterations of algorithm 1 with a phase shift, α, equal to 0.1. The number of connections is six for networks (a) and (b), twelve for (c) and (d) and 14 for (e) and (f).

5 The code.

The code can be found at: https://github.com/BotLauri/TOUDAI_PROJECT.

References

Chen, Wei et al. "Optimizing synchrony with a minimal coupling strength of coupled phase oscillators on complex networks based on desynchronous clustering". In: *Phys. Rev. E* 105 (4 Apr. 2022), p. 044302. DOI: `10.1103/PhysRevE.105.044302`. URL: `https://link.aps.org/doi/10.1103/PhysRevE.105.044302`.