

Negative Coefficients In Kuramoto Oscilators

Project Report

Tom Wiesing
Alee Kazmi

May 22, 2016

Abstract

In this project we investigate Kuramoto oscilators which were first introduced by Kuramoto in 1975. In particular we consider a situation with negative coupling coefficients and no external driver. We study patterns exhibited by these systems and design a method of constructing systems that generate desired patterns.

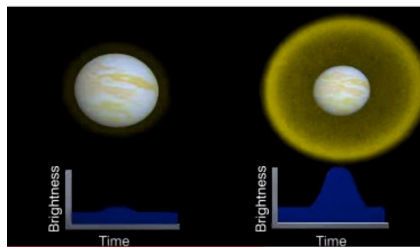
Contents

1	Introduction	3
1.1	Coupling Oscillators	3
1.2	The Firefly Model	4
1.3	Project Goal	5
2	The Kuramoto oscillator	6
2.1	The Coupling matrix and associated network topology	7
2.2	Implementation of the system	8
3	Patterns in negative coefficient systems	10
3.1	The 2-oscillator System	10
3.2	Oscillators On A Line	10
3.3	Oscillators On A Circle	13
4	Reproducing desired patterns	15
4.1	Defining Pattern Reproduction	15
4.2	Simple Patterns	16
4.3	Creating More Complex Patterns	16
5	Conclusion	19

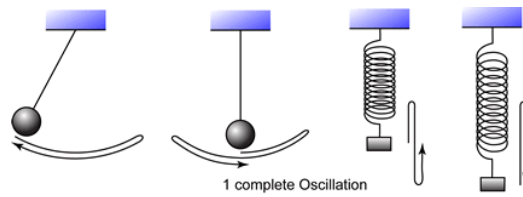
This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

1 Introduction

Oscillators can be described as a repetitive motion of some measure about a central value which is often the point of equilibrium. This term is often used in mechanical systems but it must also be noted that oscillations occur in dynamic systems too, such as economic graphs, geothermal temperatures across areas and periodic “firing” of fireflies in nature. Some examples of oscillating objects can be found in Figure 1.



(a) Cepheid Stars are known to flash occasionally. Such a flash is an example of an oscillation.



(b) A pendulum swinging to the left and right and a spring oscillating up and down.

Figure 1: Some examples of oscillating objects

1.1 Coupling Oscillators

Coupled Oscillations are a slightly more complex form of ordinary oscillators. In this model the oscillators are connected such that energy can be transferred between them. This motion can be very complex and does not have to be periodic. However, in the bigger scheme of things, every oscillator can be viewed as having a very well defined frequency of its own. Perhaps the simplest example of coupling could be a gear that transmits torque between two shafts (see Figure 2) that are not collinear.

A slightly more complex situation arises when two pendulums are joined together by an energy medium, for example a string. As we can see in Figure 3, the two pendulums have opposite amplitudes: The first pendulum only attains its maximum amplitude when the second has its lowest one. This form of synchronization behaviour can only be achieved after sufficient time has passed.

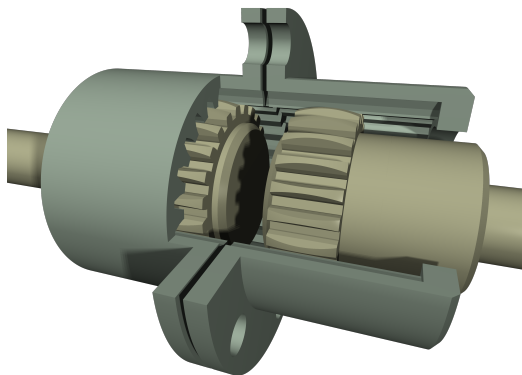


Figure 2: Image of a gear that transmits torque between two non-collinear shafts.

In any oscillating system of multiple objects synchronization can occur in two ways, (1) through coupling or (2) through randomness. In the first case some form of communication between the oscillators occurs allowing them to couple. In the second case it just so happens that they start out in a similar situation and are naturally synchronized. In this study we are only interested in the first case.

Communication can be achieved by using a medium such as a string between oscillators or by just the intrinsic tendency of natural beings to produce a unison of movement, such as fireflies flashing or humans clapping in a room.

1.2 The Firefly Model

The firefly flashing, more commonly known as the firefly model, is a classic example of synchronisation. This is modelled in detail by Kuramoto oscillators which is the main topic of study for this report. We will introduce these in the next section.

The light patterns of fireflies are part of a mating display that the male fireflies use to attract the female ones. If successful, the female replies back with a characteristic flash of its own. Although this is synchronisation on its own, there are a few species of fireflies where more than 2 individuals can synchronize. These species are usually observed in large groups, all flashing together in unison.

As we can see in Figure 4, all of the species fire off one after another in a

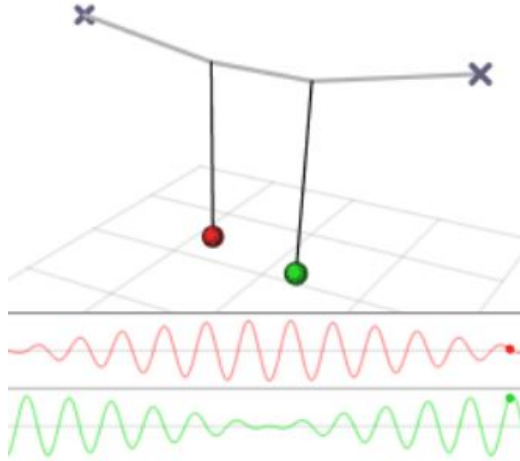


Figure 3: Two oscillating springs that exhibit coupling behaviour because they are connected through a string

synchronized fashion except the blue one which seems to be random.

1.3 Project Goal

Our goal for this study is to investigate Kuramoto oscillators with negative coupling co-efficients. We will define what exactly a coupling coefficient is and what negative coefficients mean in the next section, but one can intuitively think of it as a number measuring the strength of attraction between two oscillators. The reason we specifically want to use negative coupling co-efficients is because, in contrast to positive coefficients, it is much easier to generate complex synchronisation patterns. We want to study and observe these patterns and investigate if we can construct systems that generate desired patterns.

In Section 2 we will introduce the Kuramoto oscillator, explain its equations, some required background knowledge as well as our implementation of it. In Section 3 we continue with describing the patterns we encountered when using negative coefficients. We then move on to describe a method which can be used to re-create arbitrary patterns in Section 4 before concluding in Section 5.

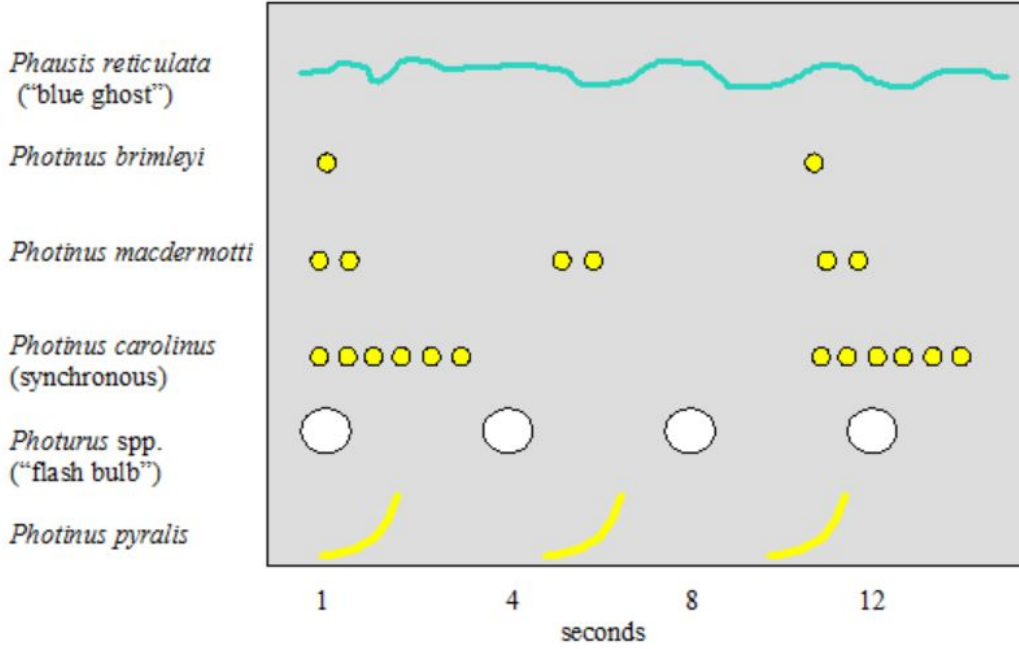


Figure 4: Behaviour of fireflies flashing over the time.

2 The Kuramoto oscillator

The problem of having a naive viewpoint on the subject of synchronization is that one may believe that every different coupling situation would require its own treatment. This may stem from the fact that one may be focusing too much on the differences rather than the similarities which are found in all synchronous systems. In this project we want to use the model of oscillators that was first introduced by Kuramoto in [1]. The most common form is given by the following system of ODEs:

$$\dot{\theta}_i = \omega_i + \sum_{j=0}^{N-1} A_{i,j} \sin(\theta_j - \theta_i) + b_i \sin(\Omega(t) - \theta_i)$$

In this equation

- θ_i is the phase of the i^{th} oscillator,
- N the number of oscillators,
- ω_i is the natural frequency of the i^{th} oscillator,

- A is the matrix of coupling coefficients of the system,
- $\Omega(t)$ is the phase of an external driver to the system and
- b is the vector of coupling coefficients to the external driver

In the Kuramoto oscillator model the main object of study is the phase of N different so-called Kuramoto oscillators, θ_0 to θ_{N-1} . It should be noted that one usually looks at this modulo 2π .

Each oscillator oscillates at its own natural frequency ω_i . However they are also influenced by the different other oscillators as well as by an external driver to the system. The observation of the other oscillators is represented by the periodic sin function. How strong each oscillator is influenced by the others and the external drivers is measured by the so-called coupling coefficients $A_{i,j}$ and b_i respectively.

2.1 The Coupling matrix and associated network topology

We can easily see from the notation above that we can consider the coupling coefficients $A_{i,j}$ as a square matrix A of size N . Similarly we can consider the coupling to the external driver as a vector b of length n .

It turns out that if we take a symmetric matrix A the system exhibits very interesting behaviour that can be related to another interpretation of A . A symmetric matrix containing only zeros and ones can naturally be seen as the adjacency matrix of an undirected, finite graph. In an adjacency matrix the elements indicate whether pairs of vertices are connected via an edge or not, i.e. if the value is 1 there is an edge, otherwise there is not. This information can then be used to reconstruct the graph from the matrix.

When solving the system this natural interpretation of A is very useful to understand the dynamics. A typical plot generated when simulating the system can be found in Figure 5. On the top is the graph that is created from an adjacency matrix A . Each vertex of the system represents a single oscillator. On the bottom is a plot of the phases of the oscillators of the system over time.

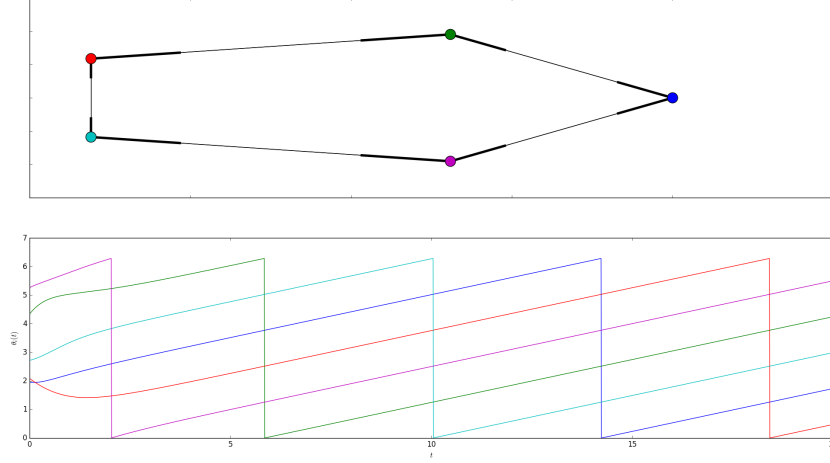


Figure 5: A typical plot of a topology and output of the system

2.2 Implementation of the system

We have implemented a simulation of this system using Python3. Our implementation uses the python packages Numpy, Scipy and Networkx. In particular it leverages the standard ODE solver functionality provided by `scipy.integrate.odeint`. Our code is very modular and consists of four distinct modules:

1. A simulator of the system, found in `oscillator.py`. This takes the parameters for the ODE and runs a simulation of the system given some initial values.
2. Some plotting code, found in `plotter.py`, which takes the output of the system and makes plots like the ones that can be found in the various Figures in this report.
3. Some code that generates adjacency matrices for specific graphs, found in `graph_generators.py`.
4. the main code of the system, found in `code.py`, which utilises the components above and runs the entire simulation.

The gist of our simulation code happens in the line below. Here `self.to_equation()` reads the current parameters and generates a function that directly represents the ODE we want to solve. To improve performance, this compiles a string of python code at runtime to a function. This is then used together with initial values `y0` and times `t` to simulate the ODE for and passed to the `odeint` function.

```
sol = odeint(self.to_equation(), y0, t, *args, **kwargs)
```

In order to interpret the matrix A as a graph, we use the `networkx` library and a slightly adapted version of the code below.

```
# create a new graph
G = nx.DiGraph()

# read current parameters
A = self.A
N = self.N

# add all the edges
for i in range(N):
    for j in range(N):
        if A[i, j] != 0:
            G.add_edge(i, j, weight=A[i,j])

# and return the graph
return G
```

Finally to generate Figures such as Figure 5, we then plot this graph together with the solution provided by `odeint`.

3 Patterns in negative coefficient systems

For our investigation of the negative coefficient situation we consider a simplified version of the Kuramoto oscillator. Specifically we assume that all oscillators have the same basic frequency ω_i , which we just call ω . We also assume that there is no external driver to the system, i.e. $b_i = 0$. This simplifies the system of ODEs to:

$$\dot{\theta}_i = \omega + \sum_{j=0}^{N-1} A_{i,j} \sin(\theta_j - \theta_i)$$

Furthermore we assume that all coefficients in the matrix A are either -1 or 0 . We can thus see A as the negative of an adjacency matrix of a graph. Here each oscillator is considered as a node of a graph and an edge can be interpreted as A having a -1 in the appropriate place.

3.1 The 2-oscillator System

The first situation we consider is the case of only 2 oscillators. It is obvious that if the oscillators are not connected through an edge, no synchronization occurs. Thus the only situation of interest is the case where they are connected through an edge, i.e.

$$A = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

We have simulated this situation with random initial values for the oscillators. As can be seen from the typical result in Figure 6 the two oscillators lock to the same frequency but have a phase shift of π , i.e. they are in anti-phase. We observed that this behaviour occurs independent of the initial values. This behaviour stems from the fact that the two oscillators are influencing each other with equal negative coefficients and thus want to oscillate as far away from each other as possible.

3.2 Oscillators On A Line

Next we want to expand this behaviour to the situation of several oscillators on a line. In this sense, that refers to the situation where N oscillators are ordered linearly and each oscillator is only connected to the next and previous

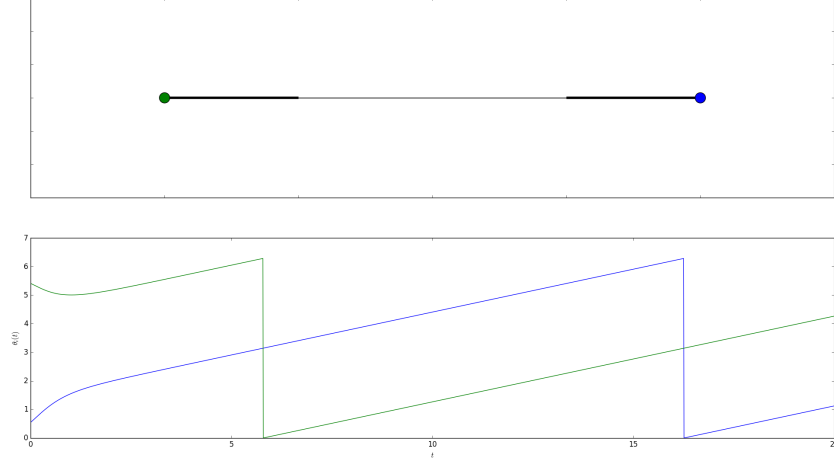


Figure 6: Simulation of a simplified 2 oscillator Kuramoto system with $\omega = 0.3$ and random initial values. A is given by the negative of the adjacency matrix of the associated graph topology which can be seen in the upper part of the figure. The behaviour of the oscillators over time can be seen in the lower part.

in the line with the exception of the ones at either end of the line. The associated matrices A to these situations are 0 except as two bands above and below the diagonal.

In the case of three oscillators the adjacency matrix is given by

$$A_3 = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

and in the case of 10 oscilators it is given by

$$A_{10} = \begin{pmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{pmatrix}$$

The typical behaviour of such systems can be seen in Figures 7 and 8 respectively. The behaviour naturally extends the behaviour of the 2 oscilator systems above: Neighbouring oscilators are in exact anti-phases. Experiments showed this behaviour to be independent of initial values and the number of oscilators, although the higher the number of oscilators, the longer the system needs to reach this state.

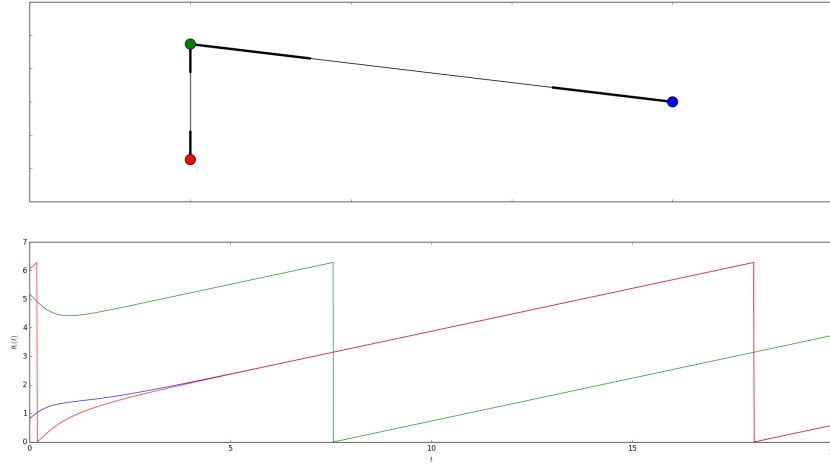


Figure 7: Simulation of a 3 oscilators-on-a-line Kuramoto oscilator system with $\omega = 0.3$ and random initial values. Again A is given by the negative of the adjacency matrix of the associated graph topology.

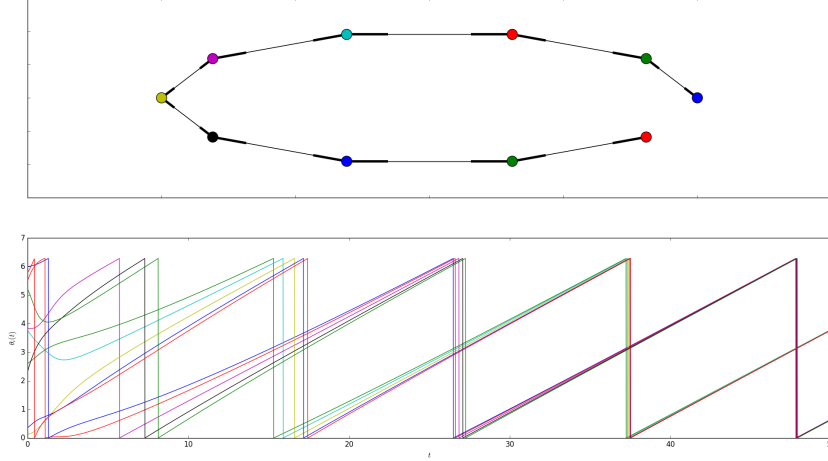


Figure 8: Simulation of a 10 oscillators-on-a-line Kuramoto oscillator system with $\omega = 0.3$ and random initial values. Again A is given by the negative of the adjacency matrix of the associated graph topology.

3.3 Oscillators On A Circle

The final patterns we observed were patterns in the case of N oscillators on a cycle. This graph looks very similar to the oscillators-on-a-line situation except that the first and last nodes are connected via an edge as well. We have again simulated the system and in this situation we came to interesting results. In the case of 3 oscillators on a cycle (which can be seen in Figure 9) the behaviour is obvious: The oscillators synchronize with phase-shifts of exactly $\frac{2\pi}{3}$.

The behaviour becomes more interesting however if we choose a different number of oscillators, for example 6. In this case we observed three different situations with three different kinds of phase shifts: Either phase shifts of π (nodes with distance 2 were exactly in phase, see Figure 10), phase shifts of $\frac{2\pi}{3}$ (nodes with distance 3 were exactly in phase, see Figure 11) or phase shifts of $\frac{2\pi}{6}$ (see Figure 12). The last of these patterns could only be produced artificially by choosing the desired situation as initial values and proved to be very unstable.

From these patterns we can deduce that the possible patterns in the general case of N oscillators-on-a-circle depends on the divisors of N . Since

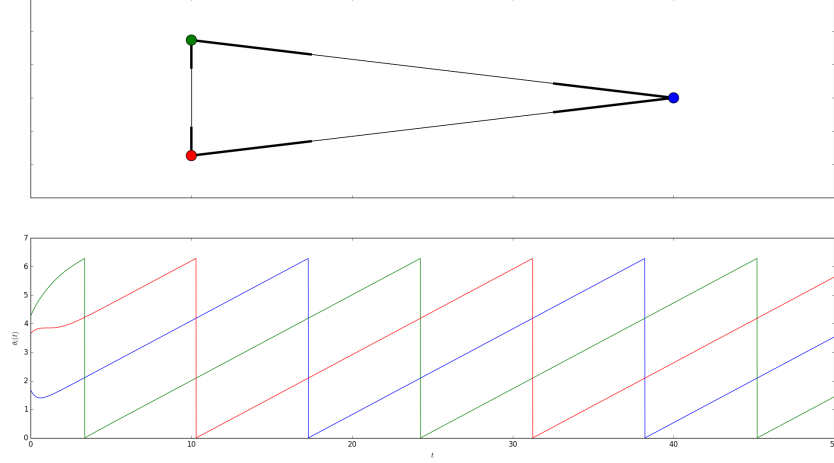


Figure 9: Simulation of a 3 oscillators-on-a-circle Kuramoto oscillator system with $\omega = 0.3$ and random initial values. Again A is given by the negative of the adjacency matrix of the associated graph topology.

each oscillator wants to be as far away from its neighbours as possible, they space out equally with respect to their phase shifts. However since each oscillator has to be in phase with itself, the sum of all phase shifts around the cycle must be a multiple of 2π . Thus for each divisor of N we get a different pattern, which is exactly what we observed.

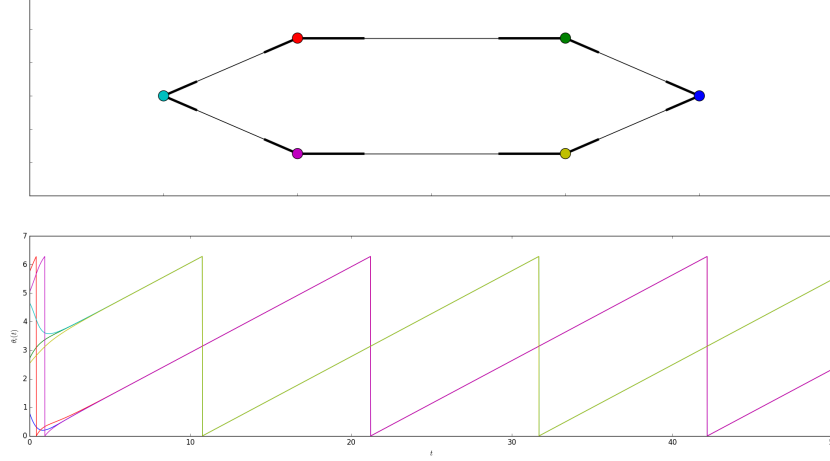


Figure 10: Simulation of a 6 oscillators-on-a-circle Kuramoto oscillator system with $\omega = 0.3$ and random initial values producing a pattern of neighbouring oscillators being exactly anti-phase

4 Reproducing desired patterns

4.1 Defining Pattern Reproduction

After making the observations in the previous section the next question that naturally occurs is if given a certain pattern, is there an associated graph topology (or associated matrix A) that is capable of reproducing it? In other words given a pattern, can we “train” a topology to create it?

In order to answer this question we first need to define “pattern” properly. We consider a pattern as the set of phase shifts between the different Kuramoto oscillators. In order to be able to observe a pattern it is necessary for the oscillators to either have the same intrinsic frequencies ω_i or lock into the same frequency. In our case we just consider the first case.

The reader should furthermore note that since we only consider phase shifts, the order of oscillators themselves is not relevant and all observation should be invariant modulo 2π .

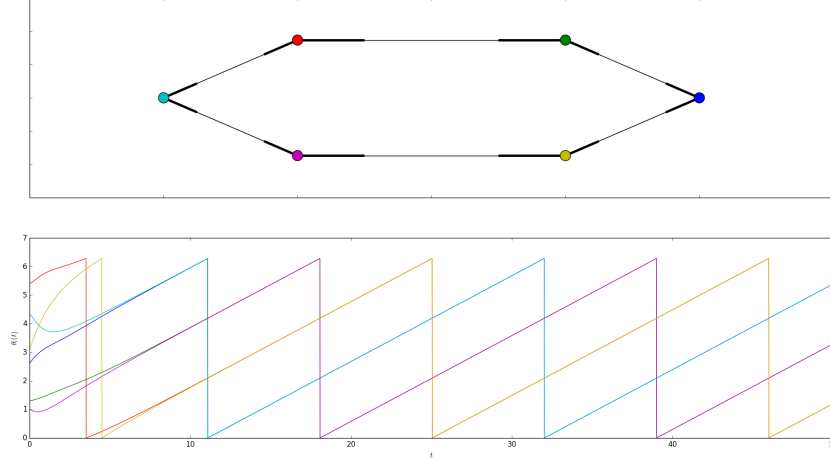


Figure 11: Simulation of a 6 oscillators-on-a-circle Kuramoto oscillator system with $\omega = 0.3$ and random initial values producing a pattern of every 3 oscillators being in sync.

4.2 Simple Patterns

We shall first consider the simple case of reproducing a simple pattern between 2 oscillators. If we take into account what we learned in the previous section, it is easily possible to re-create phase-shifts of $\frac{2\pi}{p}$ when p is prime.

If we set up p oscillators on a cyclic graph, we can get different phase shifts between neighbouring oscillators. For each divisor $d \neq 1$ of p we can achieve phase shifts of $\frac{2\pi}{d}$ between neighbouring nodes. However since p is prime, there is only a single such divisor, namely p itself, and we will always achieve a phase shifts of $\frac{2\pi}{p}$.

There is a caveat to this situation: We now have p different oscillators in our topology instead of just two. This is not a big problem however, we can just pick two oscillators at random and ignore the others.

4.3 Creating More Complex Patterns

We now have the ability to create simple patterns by setting up oscillators in a circle topology. Using the same topology we can also create offsets of the shape $\frac{a \cdot 2\pi}{p}$ where p prime and $a \in \mathbb{N}$. Instead of taking two neighbouring

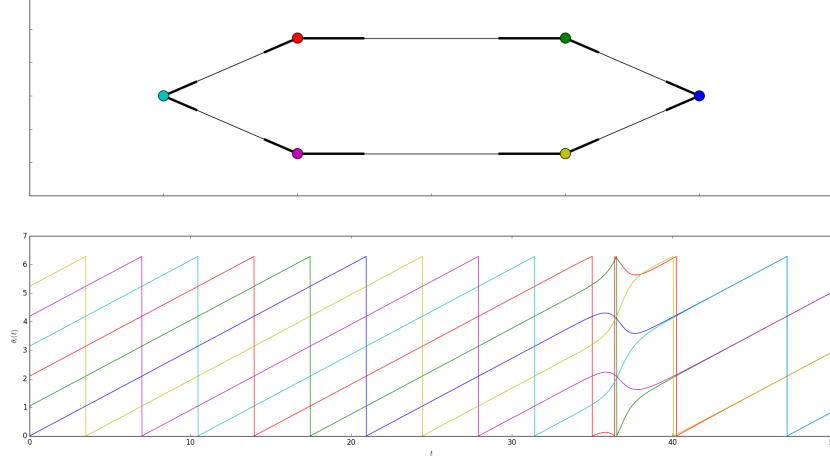


Figure 12: Simulation of a 6 oscillators-on-a-circle Kuramoto oscillator system with $\omega = 0.3$ and specific initial values producing an unstable pattern of oscillators being evenly spaced out in phase shifts.

nodes, we can just take two nodes of distance A on the same topology.

This introduces yet another caveat to the situation: We know the phase shift difference between two values but we can not predict which one is of higher value and which one is of lower value. Since we are working modulo 2π this is not even well defined.

But how can we expand this behaviour to reproduce any phase shifts $\frac{p \cdot 2\pi}{q}$ with $p, q \in \mathbb{N}, q \neq 0$? Turns out it is possible to “add” phase shifts by joining two circular topologies. Both circles will behave as they would individually and additionally the node at which they are joined will always have the same value.

If we wanted to for example reproduce the phase $\frac{8 \cdot 2\pi}{15}$ we can use one circle of size 3 and one circle of size 5. Their phase shifts of $\frac{2\pi}{5}$ and $\frac{2\pi}{3}$ add up to $\frac{8 \cdot 2\pi}{15}$, exactly what we want. The topology can be seen in Figure 13.

There is however a slight problem still: We have no guarantee that the total phase shift is indeed the sum of phase shifts in the individual circles. Since the phase shifts could propagate in different directions the total phase shift could actually be the difference. We can work around this however by choosing the correct initial values for the oscillators – both states are equally

stable.

Finally we can note that it is possible to further expand these patterns. Instead of just two circles, we can take more circles to produce offsets of the form

$$\sum_i \frac{q_i \cdot 2\pi}{p_i}$$

with p_i prime and $q_i \in \mathbb{N}$. In fact if we take into account that we are operating modulo 2π we can even take $q_i \in \mathbb{Z}$. Now given any phase shift $\frac{p \cdot 2\pi}{q}$ we can obviously use the prime factors as denominators p_i and write it in this form. We can then use the procedure described above to create a topology that can produce the given phase shift.

If we have more than two oscillators in a pattern that we want to construct, we can also continue this procedure. We can just join more circles to the node we want to define a phase shift relative to.

5 Conclusion

In the course of this project we have experimentally investigated Kuramoto oscillators with negative coefficients and developed a method to construct graph topologies that generate desired synchronisation behaviour.

Research in this topic is however far from complete. So far we have only generated experimental evidence and did not perform an analytical analysis of our approach. Furthermore our approach has some caveats, such as needing more oscillators in the generated topology than those that are involved in the pattern directly. Future work could try to optimise these solutions by using weights that are different from -1 and 0, or by using different topologies.

References

- [1] Yoshiki Kuramoto. *Lecture Notes in Physics, International Symposium on Mathematical Problems in Theoretical Physics*. Ed. by H. Araki. 1975.

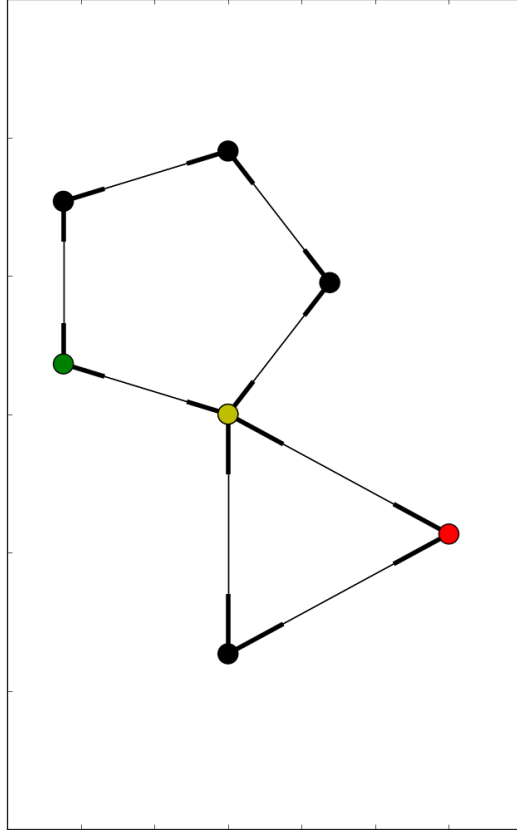


Figure 13: Graph topology used to reproduce a more complex pattern: Two circular topologies joined at the yellow node. The green and yellow nodes will have a phase shift of $\frac{2\pi}{5}$, the yellow and red nodes will have a phase shift of $\frac{2\pi}{3}$. This gives a total phase shift of $\frac{2\pi}{5} + \frac{2\pi}{5} = \frac{8 \cdot 2\pi}{15}$.