Michael Livesay, Ian Ford, Chris Gartland

Domo arigato, Mr. Kuramoto

Michael, Ian, Chris

July 25, 2014

1 Introduction

We use the Kuramoto model to study the synchronization of three oscillators with the complete graph topology. Each oscillator i is assigned a random natural frequency ω_i subject to the constraint $\sum_i \omega_i = 0$. The phase θ_i evolves according to the equation

$$\frac{d\theta_i}{dt} = \omega_i + \sum_j \gamma_{ij} \sin(\theta_j - \theta_i) \tag{1}$$

where γ_{ij} is the (i, j) entry of the adjacency matrix of the graph. This can be written in the vector form

$$\frac{d\boldsymbol{\theta}}{dt} = \boldsymbol{f}(\boldsymbol{\theta}) \tag{2}$$

for the appropriate vector field $\boldsymbol{f}$. We then plot the values of ω_0, ω_1 that result in synchronization and study the spectrum of the Jacobian matrix $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}$. Finally, we investigate variants of the Kuramoto model using periodic functions other than sine. These functions give rise to different relationships between the oscillators. We apply these variants to graphs with various topologies and study the uniqueness and nonuniqness of stable fixed points.

2 Stability Analysis of Fixed Points

For three oscillators, the vector $\boldsymbol{\omega}$ is completely determined by its first two values ω_0 and ω_1 (due to $\sum_i \omega_i = 0$), so the study of solutions to Equation 1 for varying $\boldsymbol{\omega}$ lends itself to planar visualization. To this end, we choose values for ω_0, ω_1 from -3 to 3 with a step size of 0.1 and numerically solve Equation 1 with initial condition $\boldsymbol{\theta} = \mathbf{0}$. We say that the oscillators synchronize if the Euclidean distance between $\boldsymbol{\theta}(0.8)$ and $\boldsymbol{\theta}(0.9)$ is less than 0.1 and that they diverge otherwise. Points (ω_0, ω_1) that yield synchronization are plotted as red dots and points that diverge are plotted as blue squares. Below is our implementation in Python:

```
import networkx as nx
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp

G = nx.complete_graph(3)
gamma = nx.to_numpy_matrix(G)
```

```
omega = np.zeros(len(G.nodes()))

def f(theta,t):
    N = len(theta)
    dtheta = np.zeros(N)
    for i in range(N):
        sigma = 0.
        for j in range(N):
            sigma += gamma[i,j]*np.sin(theta[j]-theta[i])
        dtheta[i] = omega[i] + sigma
    return dtheta

fig = plt.figure()
ax = fig.add_subplot(111,aspect='equal')
axis = np.linspace(-3,3,61)
t = np.linspace(0,1,11)
for x in axis:
    for y in axis:
        omega[0] = x
        omega[1] = y
        omega[2] = -x-y
        theta = odeint(f,np.zeros(3),t)
        if sp.linalg.norm(theta[-2,:]-theta[-1,:])<0.1:
            ax.plot(x,y,'ro')
        else:
            ax.plot(x,y,'bs')
plt.show()
```

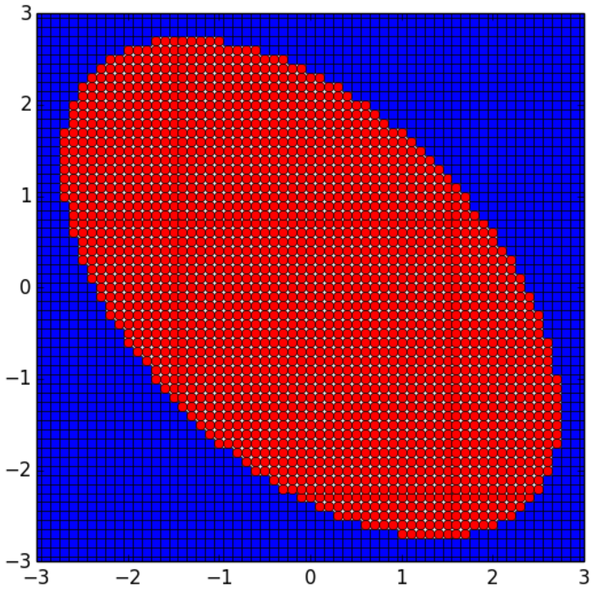


Figure 1: Points (ω_0, ω_1) belonging to the red colored region induce synchronized states, points belonging to the blue colored region induce diverging states.

This plot shows which values of $\boldsymbol{\omega}$ which yield synchronization, but gives no information about the actual values of the solution $\boldsymbol{\theta}$. One way to visualize $\boldsymbol{\theta}$ is to plot each component θ_i as a function of time on the same graph.

As an example, we choose a random $\boldsymbol{\omega} \in (-1, 1)^3$ subject to $\sum_i \omega_i = 0$ and random initial condition $\boldsymbol{\theta_0} \in [0, 2\pi)^3$, and plot the solutions in Figure 2.

```
omega = np.random.random(len(G.nodes()))
omega -= sum(omega)/len(G.nodes()) #constraint
theta0 = 2*np.pi*np.random.random(len(G.nodes()))
theta = sp.linalg.odeint(f,theta0,t)
plt.figure()
for j in range(len(G.nodes())):
    plt.plot(t,theta[:,j])
plt.show()
```

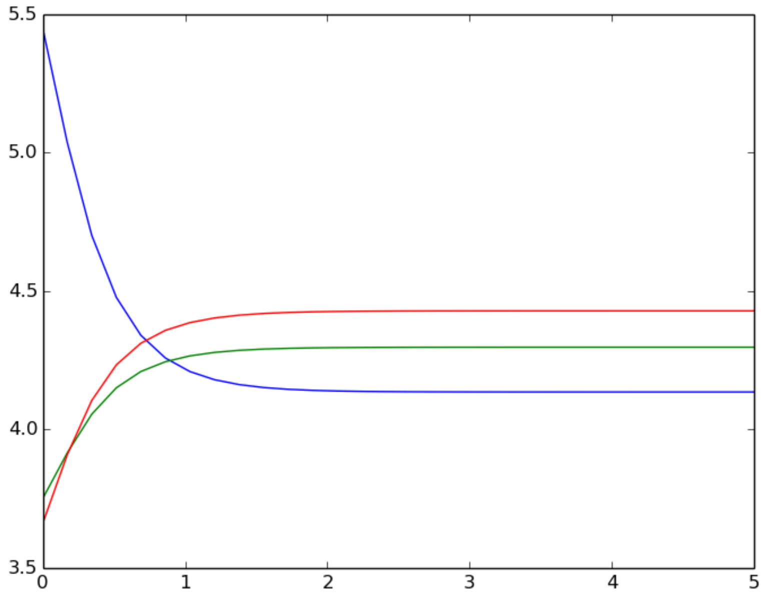


Figure 2: θ_i as a function of t for $i = 0, 1, 2$. $\boldsymbol{\omega} = (-0.45, 0.03, 0.42)$ and $\boldsymbol{\theta_0} = (5.45, 3.75, 3.66)$ for this case, and the synchronized state is $\theta = (4.14, 4.30, 4.43)$

Note that the solution converges to a synchronized state. This is due to the fact that we chose (ω_0, ω_1) in the region $[-1, 1]^2$ which is a subset of the red colored region of Figure 1. Finally, we analyze the spectrum of the Jacobian $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}}$ evaluated at the synchronized state shown in Figure 2. We compute the Jacobian exactly using

$$\frac{\partial f_i}{\partial \theta_j} = \begin{cases} -\sum_k \gamma_{ik} \cos(\theta_j - \theta_i) & i = j \\ \gamma_{ij} \cos(\theta_j - \theta_i) & i \neq j \end{cases}$$

In code:

```
def jacob(theta):
    N = len(theta)
    jacobian = np.zeros((N,N))
    for i in range(N):
        for j in range(N):
            if i==j:
                sigma = 0.
                for k in range(N):
                    sigma -= gamma[i,k]*np.cos(theta[k]-theta[i])
                jacobian[i,j] = sigma
            else:
                jacobian[i,j] = gamma[i,j]*np.cos(theta[j]-theta[i])
    return jacobian
```

The spectrum is computed by:

```
lamda, v = sp.linalg.eig(jacob(theta[-1,:]))
print lamda
print v
```

For the particular solution shown in Figure 2, the spetrum is $\{0, -2.91, -2.97\}$. Note the absence of positive eigenvalues, which is expected since such an eigenvalue would indicate an unstable synchronized state, while Figure 2 clearly demonstrates stability.In fact, as $\boldsymbol{\omega}$ approaches $\mathbf{0}$, the spectrum approachs $\{0, -3, -3\}$.

3 (Non)Uniqueness of Synchronization

For choices of $\boldsymbol{\omega}$ that yielded synchronized states, we found that a possible stable synchronized state $(\theta_0, \theta_1, \theta_2)$ is not unique in the strict sense, but $(\theta_0, \theta_1, \theta_2) - (\theta_0, \theta_0, \theta_0)$ is unique modulo 2π. Furthermore, we oberserved that these synchronized states exhibited a very strong form of stability: every solution (other than unstable fixed points) converged to a stable synchronized state. Figure 3 shows this behavior for two solutions.

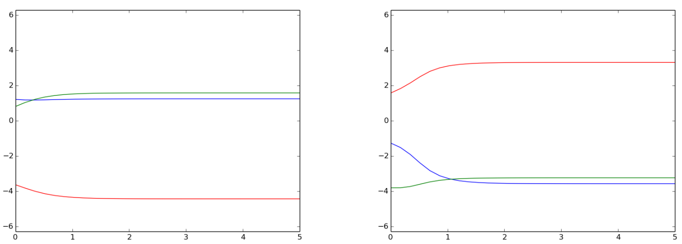


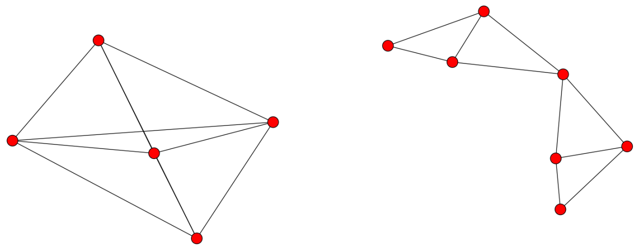
Figure 3: Solutions converging to synchronized state. These states are equivalent after translation, modulo 2π.

We experimented with different types of graph topologies and equations for θ_i to see when the elements of $\boldsymbol{\theta}$ will synchronize. Over a large set of graph topologies we found that the states of synchronization were unique when using the Kuramoto model. We found a relationship between the eigenvalues of the Laplacian of the graph topology and the strength of the synchronization of the elements of $\boldsymbol{\theta}$; for example the elements of $\boldsymbol{\theta}$ in a complete graph synchronize closer together then if the graph is not complete, when using the Kuramoto model. We were never able to find a graph topology that gave a nonunique synchronized state for our $\boldsymbol{\theta}$, using this model. In conclusion the Kuramoto model is a good way to model these graph topologies.

On the other hand we found equations which obtained unique states of synchronization for some graphs and yet had multiple states of synchronization on other graphs. Consider the following two models and the following two graphs:

$$\frac{d\theta_i}{dt} = \omega_i + \sum_j \gamma_{ij} \sin^3(\theta_j - \theta_i) \tag{3}$$

$$\frac{d\theta_i}{dt} = \omega_i - \sum_j \gamma_{ij} \sin(\theta_j - \theta_i) \tag{4}$$

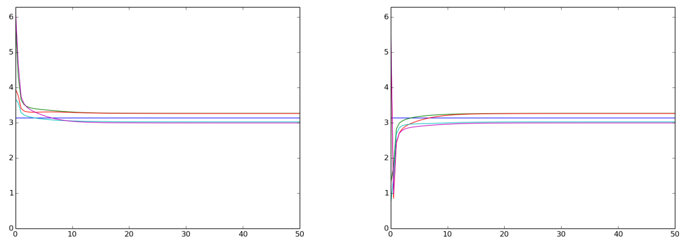


(a) Complete graph w/ 5 nodes

(b) Graph given at the beginning of the week

Figure 4: Graph Topologies

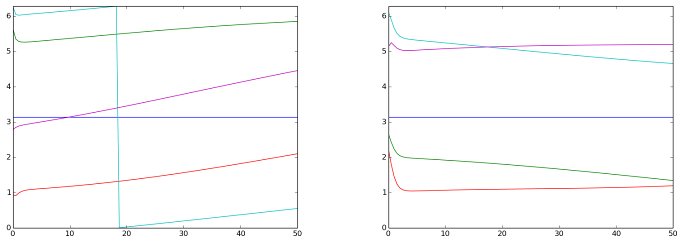
Now let us consider these methods. On the complete graph (a), equation (3) produces a stable synchronous state, but equation (4) produces slowly convergent nonunique stable synchronous states. But on the other hand in graph (b) we have the opposite result. See the following figures.



(a)

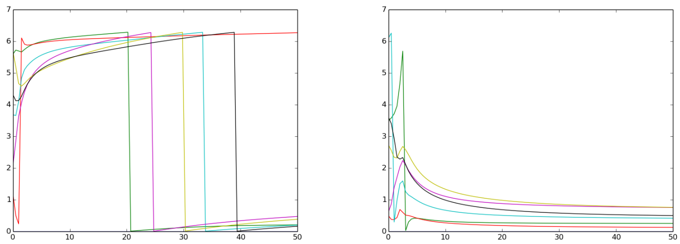
(b)

Figure 5: Uniqueness of synchronization state using equation (3) on graph (a)



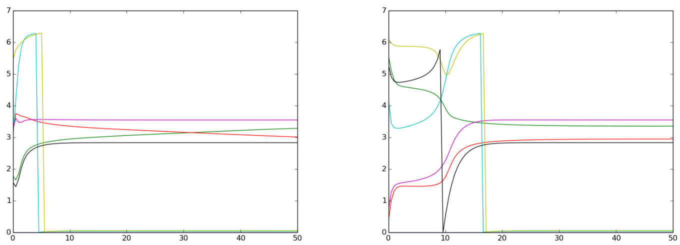
(a) (b)

Figure 6: Nonuniqueness of synchronization state using equation (4) on graph (a)



(a) (b)

Figure 7: Nonuniqueness of synchronization state using equation (3) on graph (b)



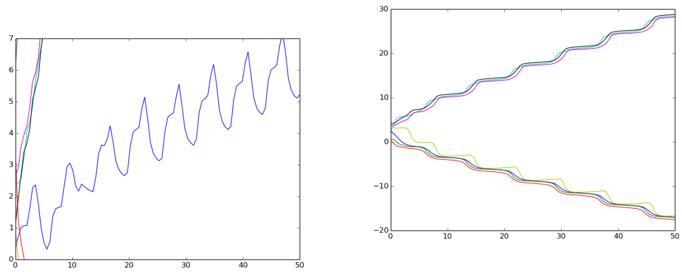
(a) (b)

Figure 8: Uniqueness of synchronization state using equation (4) on graph (b)

4 Some Other Illustrations

We have plotted a spectrum of syncronization states of the Kuramoto model on graph (b) to get a better understanding the elements of $\boldsymbol{\theta}$ move in general. Also, I've posted two plots where synchronization is essentially achieved in all but one of $\boldsymbol{\theta}$'s elements. This caused an endless motion in the other elements although their average value remained the same. This was achieved by carefully selecting the maximum ω to be between values that were sure to converge and values that were sure to never synchronize. Finally, our last figure demonstrates the behavior when an even periodic function is used in the model; in this figure we used equation (5) as our model.

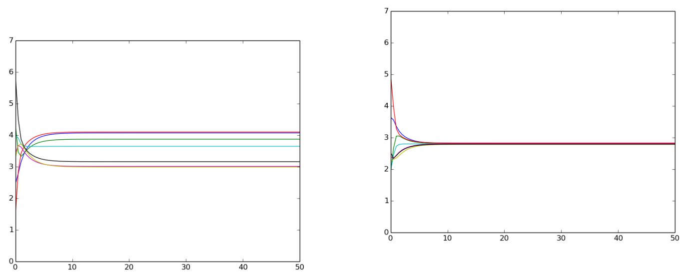
$$\frac{d\theta_i}{dt} = \omega_i + \sum_j \gamma_{ij} \tan(\theta_j - \theta_i) \tag{5}$$



(a) No synchronization (the curve in the middle is pulled by the curves above and below)

(b) Nearly synchronize (notice how the derivatives of the curves moving upward oppose the derivatives of the curves below)

Figure 9: Plots where θ never synchronizes



(a) Synchronization

(b) Strong synchronize (the elements of θ are very close here)

Figure 10: Plots where θ synchronizes

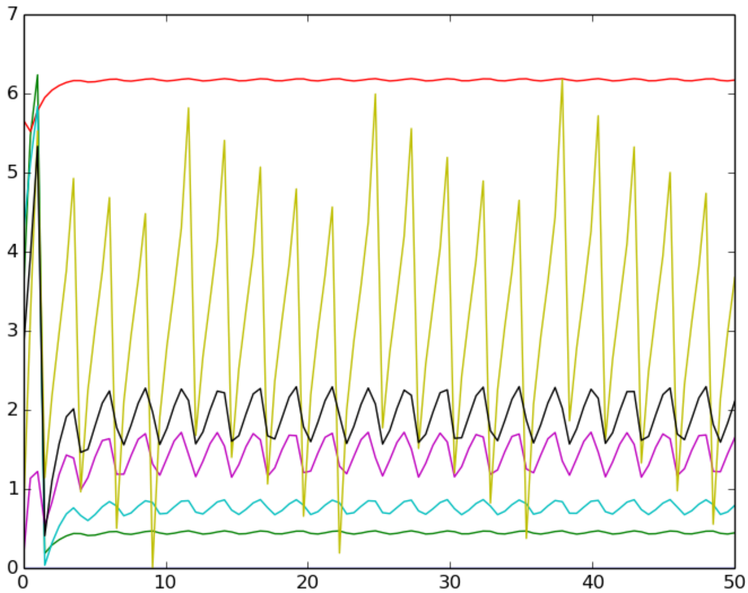


Figure 11: Seven curves bounded with respect to each other but maintaining a nonsynchronized state (w/ reference curve at frequency 60/sec)

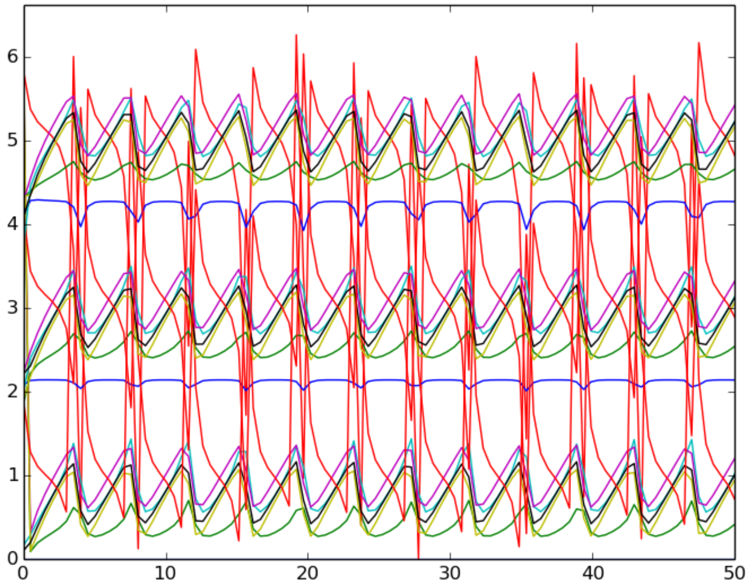
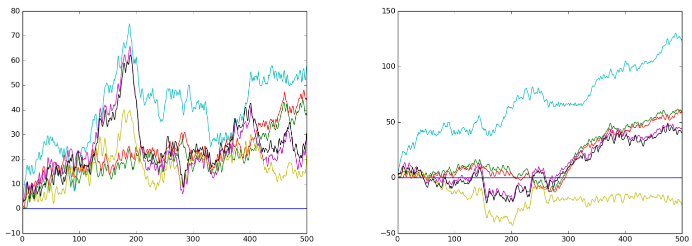


Figure 12: 3-phase bounded but out of synch (w/ reference curve at frequency 60/sec)



(a)

(b)

Figure 13: Jacobian is a skew-symmetric matrix