A PROJECT REPORT

ON

"Synchronization and Collective Dynamics of Non-Linear Systems"

Submitted to

BITS Pilani K.K. Birla Goa Campus

BY

Ishaant Agarwal	2016B5A30103G
K Pratyush Kumar	2016B5PS0435G

Under the Guidance of
Prof. Gaurav Dar

Department of Physics
BITS Pilani K.K. Birla Goa Campus

2017-2018

Acknowledgements

We are profoundly grateful to **Prof. Gaurav Dar** for his expert guidance and continuous encouragement throughout to see that this project is true to its target since its commencement to its completion.

We would like to express our deepest appreciation towards **Prof. Toby Joseph** for generously sharing his MATLAB code, that proved to be instrumental for the simulations and was one of the founding stones for this project.

At last we must express our sincere heartfelt gratitude to all the staff members and fellow students of the Physics Department who helped us directly or indirectly during the course of this project.

Ishaant Agarwal
K Pratyush Kumar

ABSTRACT

This project involved a comprehensive study of concepts related to the synchronization of dynamical systems. It also included simulations of different non-linear systems and extensive study of the synchronization behaviour as seen in the Kuramoto Model for 'n' weakly coupled oscillators.

It then focussed on a variation of the Kuramoto model with a finite number of oscillators and the study of the behaviour of systems governed by those equations. Using mathematical analysis,fixed points, corresponding to different parameters in the same model, were searched for and found, along with the occurence of some interesting bifurcations.

Keywords: Kuramoto model, synchronization, bifurcations

Contents

List of Figures

Chapter 1

Introduction

1.1 Synchronization

Synchronization is a phenomenon observed in an ensemble of oscillatory systems. It is common in biological systems : Male fireflies of *Photinus carolinus* species synchronize their flickering such that a group of fireflies will light up at and go off at the same time.[1] It has also been observed that, female members of some species who spend ample amount of time together tend to menstruate at the same time. At microscopic levels, cardiac cells and neurons exhibit synchronization. Synchronization of certain neurons leads to epilepsy, and it has been shown that it is also linked to Parkinson's disease. These systems are essentially oscillators of some form linked together by some mechanism. In light of this, study of synchronization is an area of active research not just for biologists but also for physicists and mathematicians who wish to model the oscillators in such a way that they synchronize.

One such model for coupled oscillators is the Kuramoto model [2], in our project we used the Kuramoto model of oscillators to study the conditions for which synchronization occurs.

First, we give a brief explanation of the concepts involved in nonlinear oscillators, their response to external stimuli and finally how an ensemble of oscillators behaves.

1.2 Oscillator Concepts

1.2.1 Phase of Oscillation

Oscillators can generally be described by a single parameter, say for a Simple Harmonic Oscillator the parameter is time. Similarly the trajectories of oscillators in phase space can be described by parameters. Due to the periodicity the parameter is bounded between [0,T] where T is the period after which oscillation repeats itself.

1.2.2 Limit Cycles

Limit cycles are isolated periodic trajectories in the phase space. A system which has a stable limit cycle behaves like an oscillator on the limit cycle trajectory, and around it.

1.2.3 Isochrons

Consider a system which is spiralling on to a limit cycle, it will reach the limit cycle at $t \to \infty$. However the phase of oscillation can be defined even when the system is not exactly on the limit cycle, say the initial point is y(t), it eventually reaches a point $x(t)$ on the limit cycle as $t \to \infty$. Thus, each point on the trajectory can be mapped to a point on the limit cycle. This relation is many-one, i.e multiple y's can map to a single x. So for a particular $x(t_1)$ there exist multiple $y(t_1)$, these are said to be in the same phase as x i.e t_1. In this manner each point in the phase plane can be assigned a phase.

Now, we join all the points which have the same phase by a smooth curve, this curve is called the isochron.

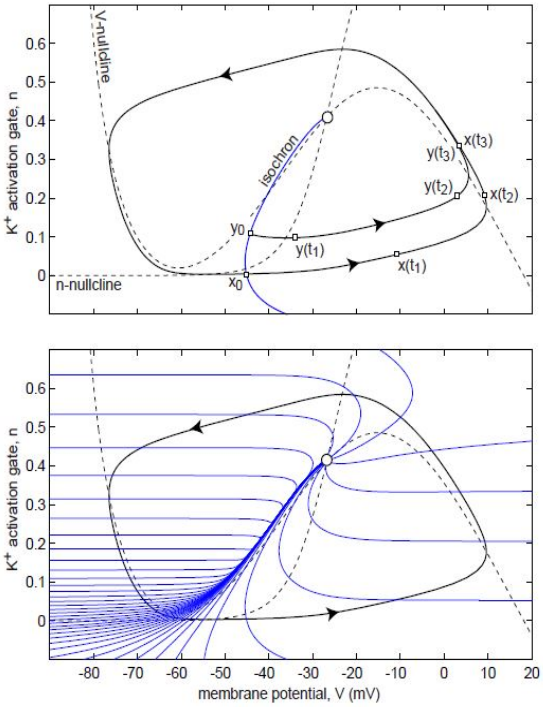

Figure 1.1: Top: An isochron, or a stable manifold, of a point x_0 on the limit cycle attractor. Bottom: Isochrons of a limit cycle attractor corresponding to 40 evenly distributed phases $nT/40$, $n = 1, ..., 40$. Reproduced from (Izhikevich 2007, Chapter 10)[3]

1.2.4 Phase Response Curves (PRCs)

When an oscillator is subjected to external stimuli in the form of discrete kicks, the phase goes into a transient state and then stabilizes to phase that is different from the originally expected phase (in the absence of kicks).

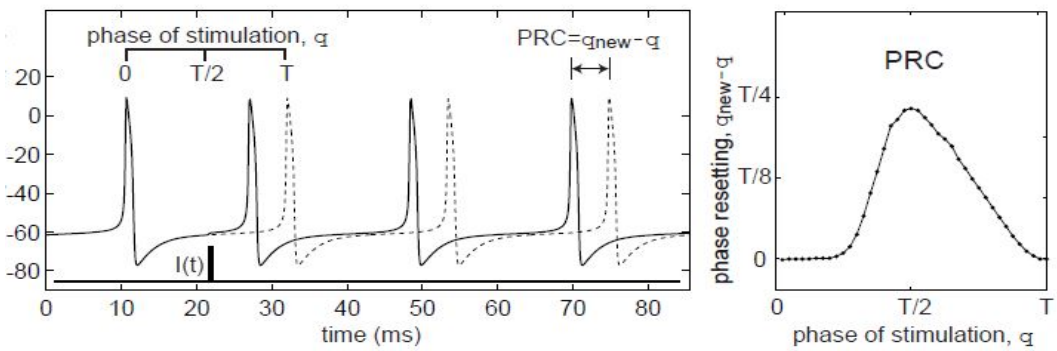

Figure 1.2: Phsae Response Curves. Reproduced from (Izhikevich 2007, Chapter 10)[3]

This change in phase depends on the form of kick and the time or phase at which the kick was applied. A Phase Response Curve is a function defined by $t_{new} - t_{old}$ for each t_{old}.

$$PRC(t_0) = t - t_0$$

Moving to the phase plane, a kick can be seen as a discrete change in one of the state while holding the other one constant, which also implies shifting from one Isochron to another thus leading to a change in phase.

1.3 Synchronization Concepts

1.3.1 Weakly coupled oscillator

$$\dot{x} = f(x) + \epsilon * p(t)$$

The above is the equation for an oscillator which experiences an external forcing which is captured by $\epsilon * p(t)$. The function $p(t)$ depends on the manner of coupling with other oscillators in the system. 'ϵ' is the magnitude of the external pulse and is small for weak coupling. Now if $\epsilon \to 0$ then $\dot{x} = f(x)$, which is just the equation for a self sustained oscillator whose equation can be converted into a parametric equation with time as its phase.

$$\dot{q} = 1$$

That is, the frequency has been taken as unity.

To get a similar equation for a forced oscillator Arthur Winfree, Yoshiki Kuramoto and Ioel Malkin independently developed three methods of the same name respectively.

1.3.2 Winfree's Approach [4]

Consider a well magnified neighborhood around a point on the phase plane, the isochrons passing through that point will be parallel to the isochrons that have a small phase difference.

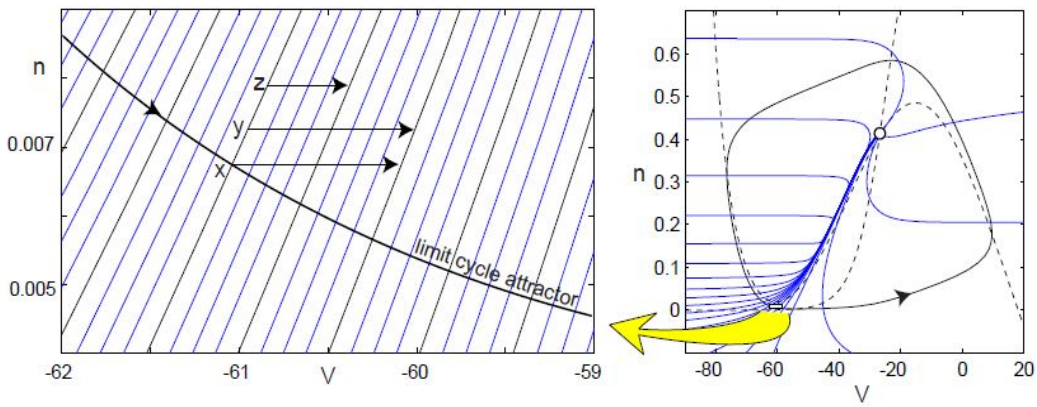

Figure 1.3: Magnified view of Isochrons. Reproduced from (Izhikevich 2007, Chapter 10)[3]

From the figure it is clear that the phase resetting is linear, that is the change in the phase is directly proportional to the magnitude of the kick or 'e' in this case. Therefore the $PRC(q, A)$ is a product of A and some function of phase to ensure linearity w.r.t A.

$$PRC(q, A) = Z(q) * A$$

We know that

$$q_{new} = q_{old} + PRC(q_{old}).$$

Instead of using a continuous function $p(t)$, it can approximated by a discrete function $p(t_n)$ which excites by an amount $\epsilon * p(t_n) * h$ for an interval h, after which the next pulse arrives.

$$h = t_{n+1} - t_n$$
$$q(t_{n+1}) = q(t_n) + PRC(q_{old}, A) + h$$
$$q(t_{n+1}) = q(t_n) + Z(q(t_n)) * \epsilon * p(t_n) * h + h$$

Rearranging,

$$\frac{q(t_{n+1}) - q(t_n)}{h} = Z(q(t_n)) * \epsilon * p(t_n) + 1$$

where the LHS is the discrete version of the derivative of q.

$$\dot{q} = 1 + \epsilon * p(t) * Z(q(t))$$

Y. Kuramoto also derived the same equation using a different approach, using gradient of the vector field etc.

1.3.3 Phase Model for Coupled Oscillators [3]

We consider a model for an ensemble of coupled oscillators

$$\dot{x}_i = f_i(x_i) + \epsilon \sum_{i,j} g_{ij}(x_i, x_j)$$

Using Winfree's Approach we get

$$\dot{\theta} = 1 + \epsilon Q_i(\theta_i) \sum_{i,j} g_{ij}(x_i, x_j)$$

The equation can be integrated to write :

$$\theta_i = t + \varphi_i$$

where $\dot{\varphi} = \epsilon Q_i(t + \varphi_i) \sum_{i,j} g_{ij}(x_i(t + \varphi_i), x_j(t + \varphi_i))$

The classical averaging theory described in Hoppensteadt and Izhikevich (1997) [5] is

used to transform the system into :

$$\dot{\varphi} = \epsilon\omega_i + \epsilon \sum_{i \neq j} H_{ij}(\varphi_j - \varphi_i)$$

$$H_{ij}(\varphi_j - \varphi_i) = \frac{1}{T} \int_0^T Q_i(t) \cdot g_{ij}(x_i(t), x_j(t + \varphi_j - \varphi_i))dt$$

Y. Kuramoto approximated the H function using the first term of Fourier Sine series and introduced a new parameter for scaling time, $\tau = \epsilon t$. This equation is called the Kuramoto Phase Model [2].

$$\psi' = \omega_i + \sum_{i \neq j} c_{ij} sin(\varphi_j - \varphi_i + \psi_{ij})$$

The constants c_{ij} are taken to be K/n, coupling constant divided by number of oscillators. The constant phase ψ_{ij} can be taken to be zero to simplify the model further. We used this model to simulate oscillators on MATLAB and its results are summarised in the following section.

Chapter 2

Results

Using the final Kuramoto model for oscillators we did simulations on MATLAB for an arbitrary number of coupled oscillators.

2.1 Simulation 1 :

For any prescribed number of oscillators, the time evolution of the system of ODEs was calculated using the Euler Method and phase difference of jth oscillator with the 1st oscillator was plotted as a function of time. The parameters that could be varied were the coupling constant and the frequency of each oscillator. A random function was used to provide initial states as well as the frequencies.

We observed that for a given set of frequencies, the oscillators sync after a critical value of the coupling constant, k.

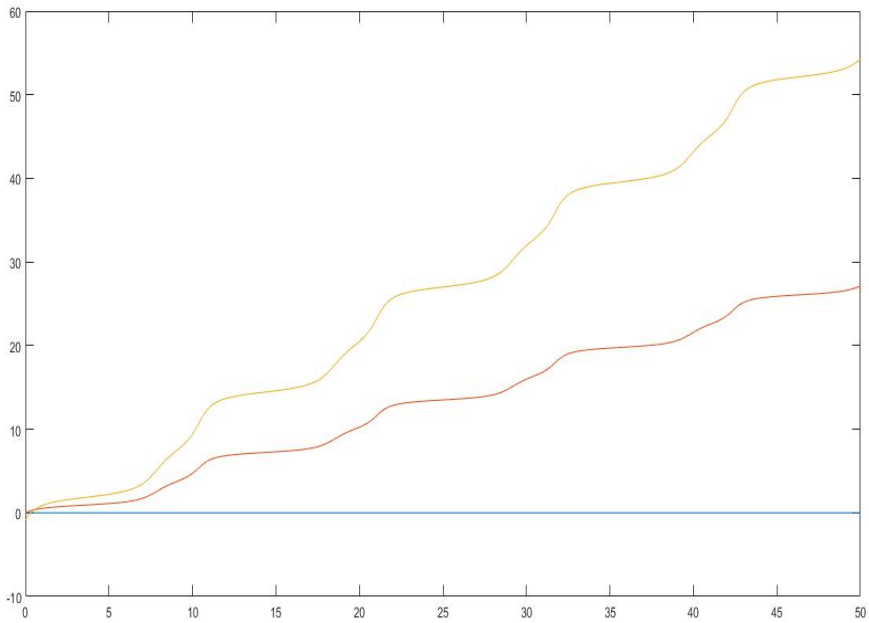

Figure 2.1: Plot of phase differences of 3 oscillators with time for k = 0.5. Note that the system is not in sync

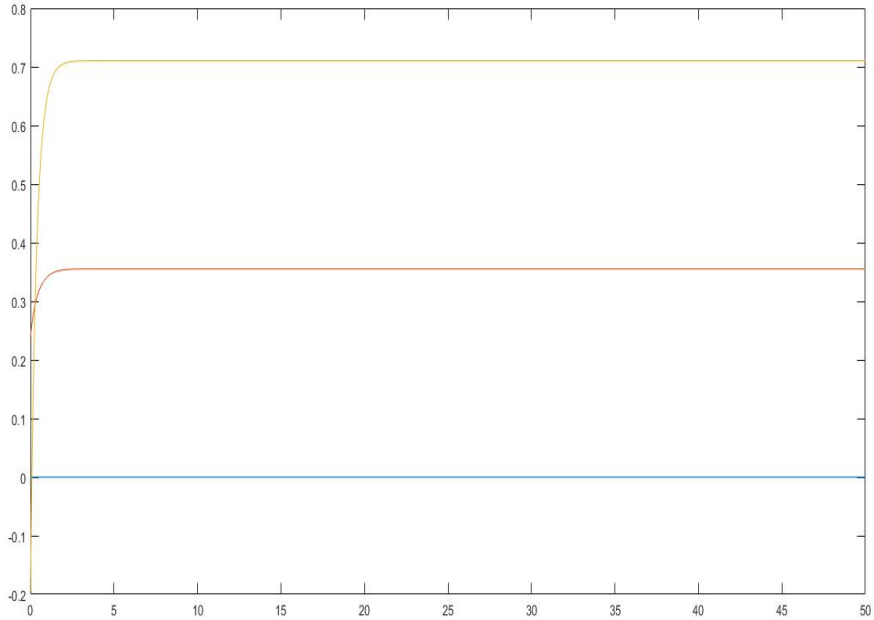

Figure 2.2: Plot of phase differences of 3 oscillators with time for k =1. The system shows synchronization

2.2 Simulation 2 :

For a better understanding the dynamics of the system and stability of fixed points of the system we wrote a code for plotting the nullclines for a system of 3 oscillators i.e for a system of 2 ODEs. The frequencies of the oscillators were kept fixed at 1, 2 and 3 while the coupling constant was changed.

We observed that for small values of 'k' the nullclines were closed loops and there were several of them in the phase plane repeating after intervals of 2π . So we confined our phase plane to -2π to 2π and it was observed that one of the nullclines starts expanding as the value of 'k' was increased and eventually the 2 loops of the same nullcline combine and branch out into 2 branches while the nullcline of the other equation follows the same suite but for even larger values of 'k'. Now this transformation of nullclines led to creation of several new fixed points which were unstable.

One of the bifurcations observed was a very peculiar one, it involved creation of an unstable fixed point at a critical value of 'k' after which it separates into 2 unstable fixed points. This bifurcation doesn't fall into any standard categorie of bifurcations. The fixed points thus created were cross checked from the ones obtained from the Newton-Raphson code and their stability was verified using Simulation 1. Another bifurcation was similar to a pitchfork, one unstable fixed point gave way to 3 unstable fixed points.

The list of critical values of k for which bifurcations occur are given in Appendix E.

The code was modified to create vector plot on the phase plane. We couldn't find any possibility of limit cycles. The frequencies of the oscillators were made unity to check the

symmetry in the phase plane, and its variation with coupling constant. It was observed that there are 4 fixed points, of which one is stable (0,0), and the phase plane remains so for all values of coupling constant.

For a negative value of coupling constant, and all frequencies equal to unity the fixed points simply changed their stability.

The following are the results obtained from the simulations.

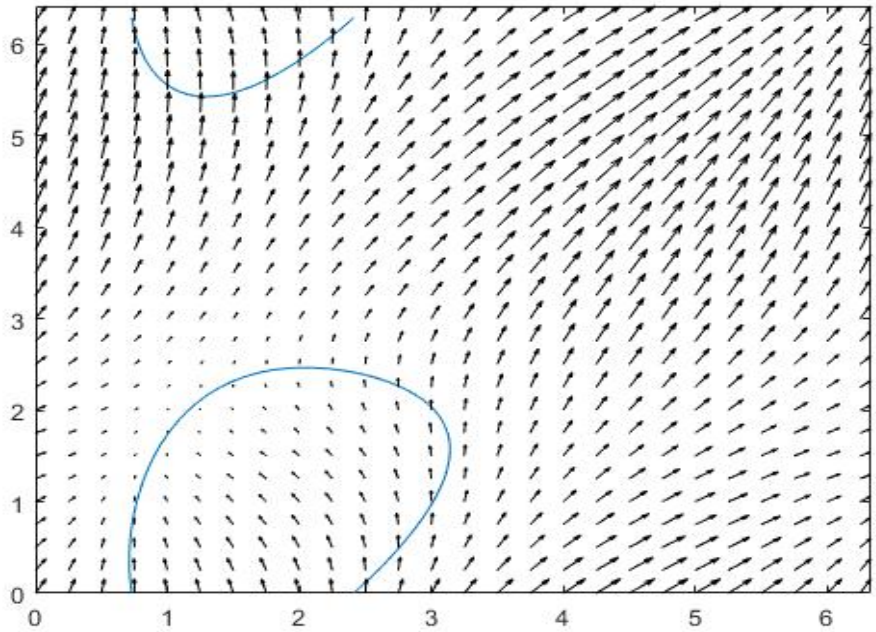

Figure 2.3: Nullclines and vector plot for a system of three oscillators with k = 0.5

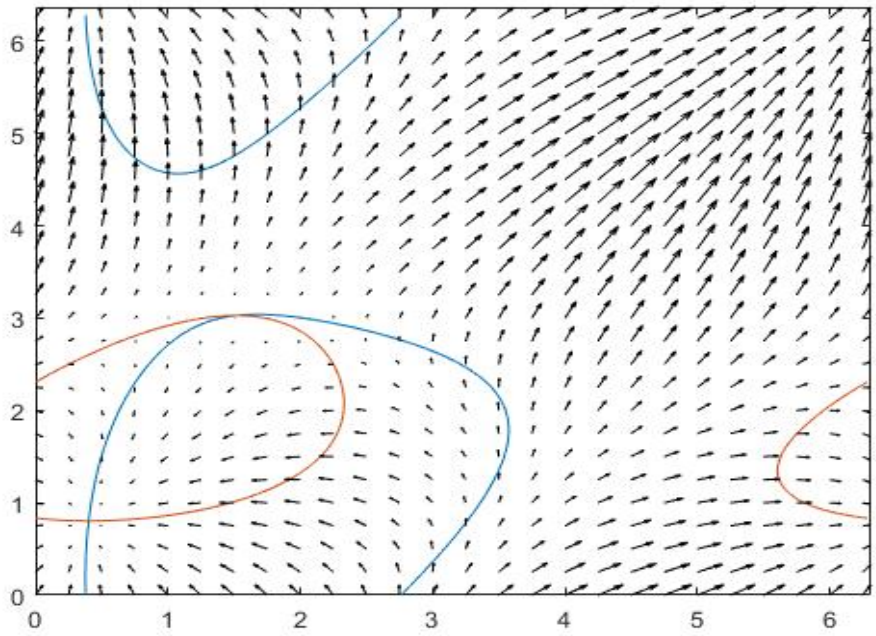

Figure 2.4: Two fixed points are seen for k = 0.9

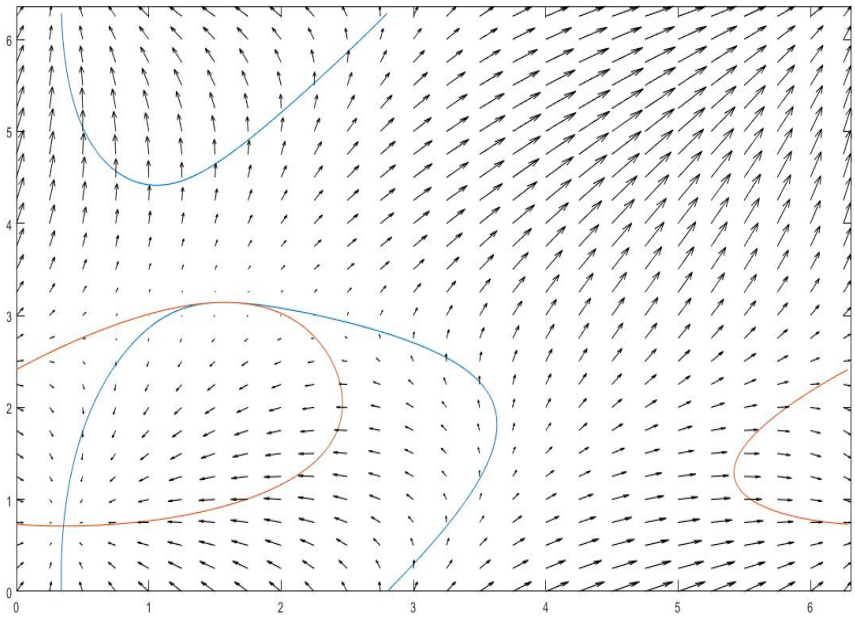

Figure 2.5: k = 1

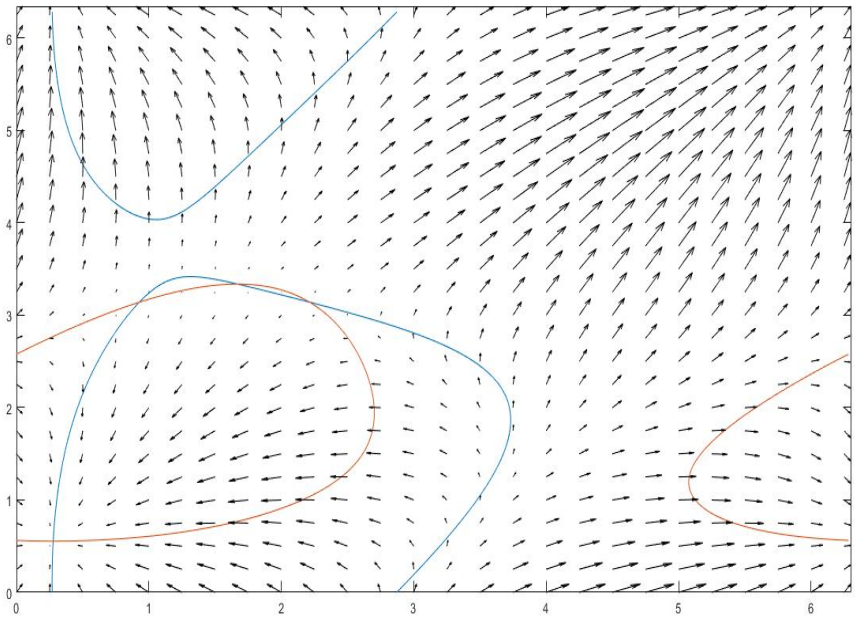

Figure 2.6: The unstable fixed point bifurcates into 3 unstable points for k = 1.25

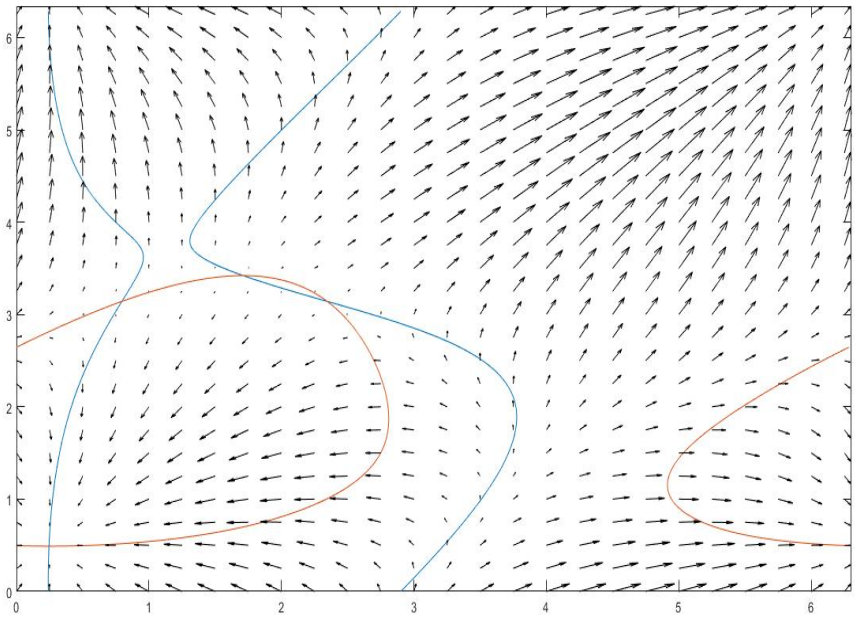

Figure 2.7: k = 1.4

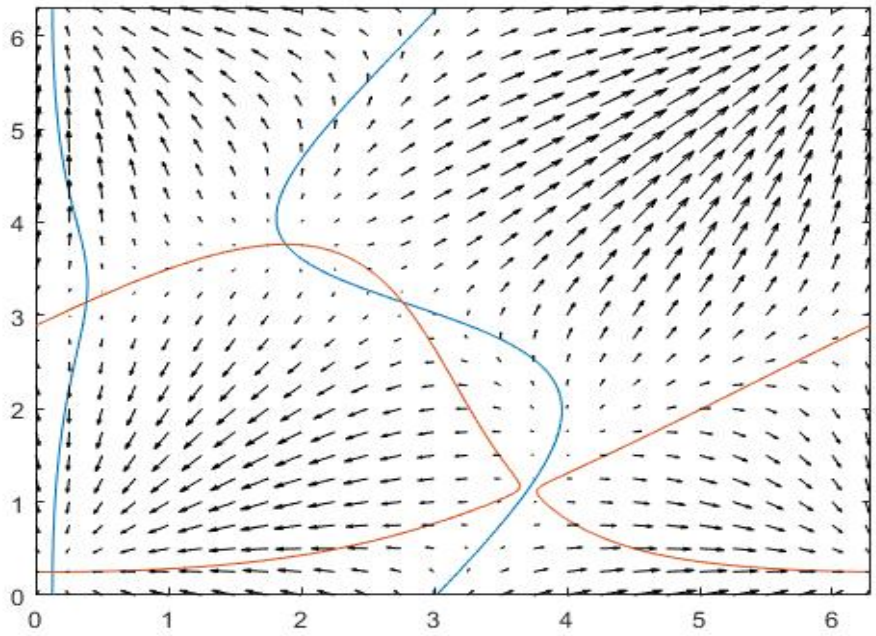

Figure 2.8: A new fixed point is about to emerge at approx. (4,1) for k = 2.7

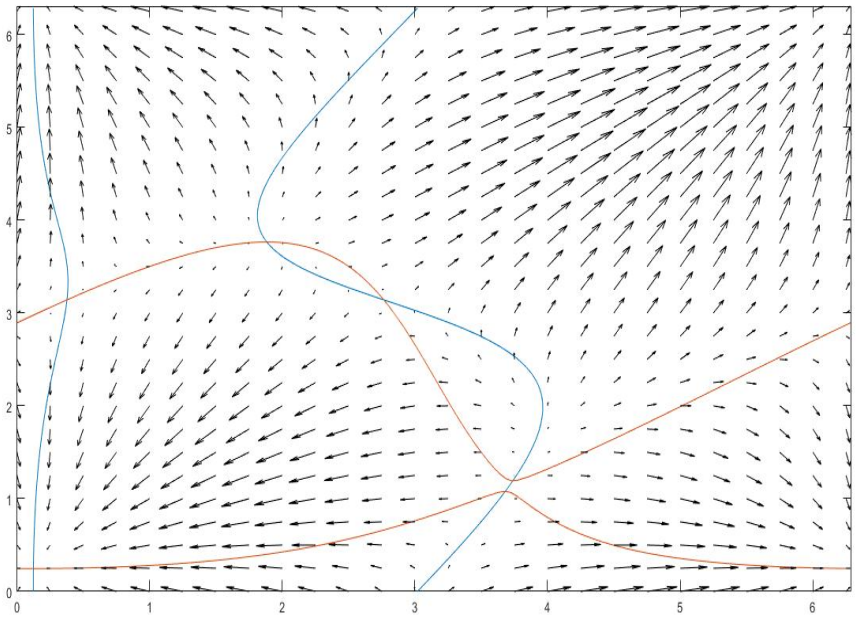

Figure 2.9: The newly created fixed point gives rise to two unstable points for k = 2.72

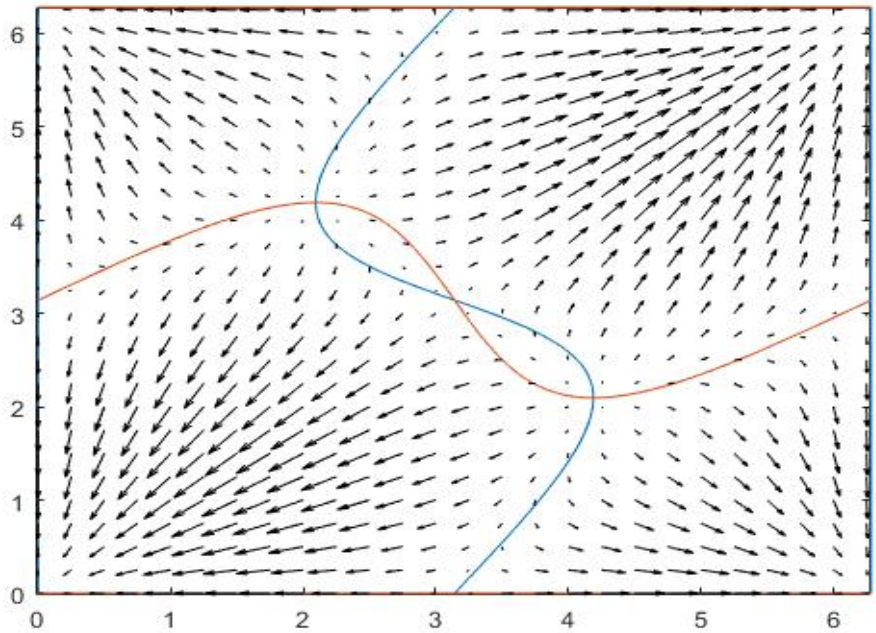

Figure 2.10: k = 0.5, all frequencies are equal. The phase plane does not show any variation with k as expected

2.3 Simulation 3 :

To find the stable and unstable states or fixed points of the synchronized oscillators the system of ODEs has to be solved simultaneously as, the Euler method can only provide the stable states of the oscillators.

The system of n-equations was reduced to system of $n-1$ equations by subtracting the phase of 1st oscillator from the remaining $n-1$ oscillators

This system of $n-1$ equations was then solved using the Newton-Raphson Method. The program was reinitialized for different initial guesses and multiple roots were observed for suitable values of the coupling constant, k. Some of these roots corresponded with the values of phase differences seen in Simulation 1 after synchronization was achieved.

Chapter 3

Conclusion and Future Scope

3.1 Conclusion

In conclusion we simulated the Kuramoto model for several oscillators, and later for 3 coupled oscillators we found the conditions on the coupling parameter for which sync is achieved and we also found a very peculiar bifurcation, which we couldn't find in any standard texts of Nonlinear Dynamics.

3.2 Future Scope

The bifurcation observed here needs to studied further and a robust mathematical description for the same is necessary.

Our study was restricted to the set of frequencies [1, 2, 3] and [1, 1, 1] for three oscillators and the behaviour for other frequencies needs to studied

References

[1] J. B. Buck, Synchronous Rhythmic Flashing of Fireflies, The Quarterly Review of Biology 13 (3) (1938) 301–314. arXiv:https://doi.org/10.1086/394562, doi:10.1086/394562.

[2] Y. Kuramoto, Self-entrainment of a population of coupled non-linear oscillators, International Symposium on Mathematical Problems in Theoretical Physics (1975) 420–422doi:10.1007/BFb0013365.

[3] E. M. Izhikevich, Dynamical Systems in Neuroscience : The Geometry of Excitability and Bursting, Cambridge, Mass: MIT Press., 2007.
URL https://www.izhikevich.org/publications/dsn/index.htm

[4] A. T. Winfree, Biological rhythms and the behavior of populations of coupled oscillators, Journal of Theoretical Biology 16 (1) (1967) 15 – 42. doi:https://doi.org/10.1016/0022-5193(67)90051-3.
URL http://www.sciencedirect.com/science/article/pii/0022519367900513

[5] F. C. Hoppensteadt, E. M. Izhikevich, Weakly connected neural networks, New York : Springer, 1997.
URL https://www.springer.com/in/book/9780387949482

[6] G. Davidowitz, H. F. Nijhout, The physiological basis of reaction norms: The interaction among growth rate, the duration of growth and body size1, Integrative and Comparative Biology 44 (6) (2004) 443–449. arXiv:/oup/backfile/content_public/journal/icb/44/6/10.1093/icb/44.6.443/2/i1540-7063-044-06-0443.pdf, doi:10.1093/icb/44.6.443.
URL http://dx.doi.org/10.1093/icb/44.6.443

Appendices

Appendix A

Simulation 1

Kuramoto_model.m

```
clear;
h=0.01;
t=0:h:50;
prompt='Number of osc : ';
%n=input(prompt);
n=3;
c=3;
%c=input(prompt);
C=c/n; %coupling constant
%%%%%%%%
C=4.5;

for(i=1:n)
    X(i,1)=normrnd(3.14,1);
    V(i,1)=normrnd(3,1.2);
end
C=1;
%X=[0 ; 4.011; 1.738];
V=[1; 2; 3];

for i=1:(length(t)-1)
    k=osc(n,C,V,X(:,i));
    X(:,i+1)=X(:,i)+k*h;
end

for i=1:n
    Y(i,:)=X(i,:)-X(1,:);
end
Z=mod(Y,2*pi);
Z(:,length(t))
plot(t,Y);

disp('End values');
Z(:,length(t))
```

code/Kuramoto_model.m

Appendix B

Simulation 2

Kuramoto_nullcline.m

```
h=0.01;
t=0:h:20;
%phi=[a; b; c];
v=[1; 2; 3];
k=3;
%k=2.709954
%psi=[phi(2,1)-phi(1,1); phi(3,1)-phi(1,1)];
%w=[v(2,1)-v(1,1); v(3,1)-v(2,1)];
limit1=0;
limit2=2*pi;
%for(i=1:length(t)-1)

f=@(psi21,psi31) v(2)-v(1) +k*(-2*sin(psi21) - sin(psi31) + sin(psi31-psi21));
g=@(psi21,psi31) v(3)-v(1) +k*(-2*sin(psi31) - sin(psi21) + sin(psi21-psi31));
%blue : psi21dot=0
figure
fimplicit(f,'-' , [limit1 limit2 limit1 limit2]);
%title('Combine Plots')
hold on
fimplicit(g, [limit1 limit2 limit1 limit2]);
%plot(x,y2)
%{
x21=0:0.05:2*pi;
x31=0:0.05:2*pi;
for i=x21
    for j=x31
        if(f(i,j)<(10^-50) && g(i,j)<(10^-50))
            disp("Fixed point : ");
            disp(i);
            disp(j);
        %else
            %disp('Not found');
        end
    end
end
%}
```

code/Kuramoto_nullcline.m

Kuramoto_vector_plot.m

```
clear all;
h=0.25;
t=0:h:20;
%phi=[a; b; c];
```

```
dpsi31=[1; 2; 3];
k=0.8;
%psi=[phi(2,1)-phi(1,1); phi(3,1)-phi(1,1)];
%w=[v(2,1)-v(1,1); v(3,1)-v(2,1)];
limit1=0;
limit2=2*pi;

f=@(psi21,psi31) dpsi31(2)-dpsi31(1) +k*(-2*sin(psi21) - sin(psi31) + sin(psi31-psi21));
g=@(psi21,psi31) dpsi31(3)-dpsi31(1) +k*(-2*sin(psi31) - sin(psi21) + sin(psi21-psi31));

%{
for i=1:length(X)
    for j=1:length(X)
        dpsi21(i,j)=f(X(1,i),X(2,j));
        dpsi31(i,j)=g(X(1,i),X(2,j));
    end
end
%}

[psi21,psi31] = meshgrid(limit1:h:limit2,limit1:h:limit2);
dpsi21 = f(psi21,psi31);
dpsi31 = g(psi21,psi31);
scale=1;
figure
fimplicit(f,'-' , [limit1 limit2 limit1 limit2]);
hold on
fimplicit(g, [limit1 limit2 limit1 limit2]);
hold on
quiver(psi21,psi31,dpsi21,dpsi31,scale,'magenta');
```

code/Kuramoto_vector_plot.m

Appendix C

Simulation 3

J.m

```
function [J] = J(~)
%JACOB computes the Jacobian matrix at x

global N;
global K;
global psi;

J= zeros(N-1,N-1);
for i=2:N
    for k=2:N
J(i-1,k-1) = (K/N)*(cos(psi(k,1)-psi(i,1))-cos(psi(k,1)));

    end
end
for i=2:N
    J(i-1,i-1) = (K/N);
    for k=1:N
    J(i-1,i-1) = J(i-1,i-1) - (K/N)*(cos(psi(k,1)-psi(i,1)));
    end
end
end
```

code/J.m

F.m

```
function [f] = F(~)
%MFUN calculates and returns the negative of the 'function vector'
%evaluated at x

global N;
global Omega;
global K;
global psi;

f= zeros(N-1,1);

for i=2:N
    f(i-1,1) = (Omega(i,1)-Omega(1,1));
    for j=1:N
    f(i-1,1) = f(i-1,1)+ (K/N)*(sin(psi(j,1)-psi(i,1))-sin(psi(j,1)));
    end
end
end
```

code/F.m

NR_Multi.m

```
% Function for finding root in multidimensional case using Newton-Raphson
% method. The code should work when the initial guess is close to one of
% the roots.
function [Root, Count] = NR_Multi(X_initial, Tol, Fun, Jacob)

Error = 10*Tol;   %Initializing Error variable to be larger than Tol
Count = 0;
%Initializing the Count variable
%delta = 100;
%Count_max = 200;   %Maximum value of Count beyond which search is terminated
global psi;
global N;

while Error > Tol
            Count = Count + 1;
        dX = linsolve(Jacob(X_initial),-1*Fun(X_initial));
        X_new = X_initial + dX;
        Error = norm(dX)/norm(X_new);
        X_initial = X_new;
    for i=2:N
        psi(i,1) = X_new(i-1,1);
    end
    %delta = norm(F(psi));

end

Root = X_initial; %Storing the final value of the root

end
```

code/NR_Multi.m

Main.m

```
clf, clear, close all
clear all

global N;
N=3;   % num of oscillators

global K;

K=9;    % Coupling strength.



global psi;

psi=zeros(N,1);

global Omega;
%Omega=rand(N,1); % Random Omegas

Omega = [1;2;3]; % Predefined Omegas

%   theta=rand(3,1);
%   theta(:,1)=[1;2;3];
%
%           for i=1:N
%                   psi(i,1)=theta(i,1)-theta(1,1);       %Psis constant
%
%           end

```

```
Root = zeros(50,2);
for l = 1:200
     theta=zeros(N,1);
     theta(:,1)=2*pi*rand(N,1);
         for i=1:N
             for j=1:N
                 psi(i,j)=theta(i,1)-theta(j,1); %Psi(s) randomised in each loop
             end
         end
 [root, count] = NR_Multi([psi(2,1); psi(3,1)], 0.00001,@F,@J);
%[root, count] = NR_Multi([ 0.7215; 1.6821], 0.00001,@F,@J);
Root(l,1)= mod(root(1,1),2*pi);
Root(l,2)= mod(root(2,1),2*pi);
end
C = unique(Root,'rows')
```

code/Main.m

Appendix D

Critical values of k

K	Stable	Unstable	Unstable	Unstable	Unstable	Unstable
1	0.335	1.6				
	0.71	3.142				
1.05	0.335	1.258	1.599	1.883		
	0.67	3.141	3.189	3.141		
1.1	0.311	1.161	1.615	2.01		
	0.6379	3.148	3.232	3.136		
1.3	0.2568	0.8742	1.667	2.266		
	0.5314	3.14	3.368	3.14		
1.35	0.2513	0.8342	1.696	2.304		
	0.5103	3.142	3.396	3.145		
1.36	0.2513	0.8263	1.676	2.304		
	0.5063	3.142	3.41	3.153		
1.9	0.1676	0.5445	1.801	2.587		
	0.3566	3.136	3.605	3.142		
2.7	0.1257	0.3794	1.885	2.762		
	0.2488	3.142	3.763	3.142		
2.71	0.1243	0.3779	1.885	2.764	3.708	
	0.24	3.142	3.764	3.142	1.139	
2.8	0.1257	0.3653	1.885	2.765	3.793	3.602
	0.2398	3.142	3.776	3.162	1.309	0.9578
4.5	0.0837	0.2241	1.969	2.917	4.021	3.393
	0.1486	3.142	3.921	3.142	1.738	0.4755