МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

Национальный исследовательский нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Кафедра теории управления и динамики систем

Направление подготовки: «Прикладная математика и информатика»

Отчет по учебной практике

Тема :

Изучение процесса синхронизации ансамбля мобильных агентов в трехмерном пространстве

Выполнил:

студент ИИТММ гр. 381903-2

Диженин Владислав Евгеньевич

Научный руководитель:

заведующий кафедры ТУДС, ИИТММ

Осипов Григорий Владимирович

Нижний Новгород

2023

Оглавление

Глава 1

Введение

Открытие сложного непредсказуемого поведения траекторий в системах - динамического хаоса, по праву считается одним из самых значимых достижений науки, благодоря которому многие проблемы техники, естествознания получили адекватное математическое описание. Неотъемлемой частью разговора о хаотических системах является понятие синхронизации.

Синхронизация является универсальным и фундаментальным природным механизмом, проявления которого можно встретить во множестве областей, такиих как биология, химия, физика, техника и т.д.

Именно такое широкое распространение явления снхронизации подталкивает исследователей к изучению этого феномена, оно и послужило причиной зарождения отдельного раздела в теории нелинейных колебаний - теории синзронизации.

Более актуальным примером будет следующее применение синхронизации. В связи с последними мировыми тенденциями и активным развитием малых беспилотных летательных аппаратов, появилась необходимость в изучении и реализации управления большими группами устройств такого класса. Данную задачу может решить процесс синхронизации движущихся объектов. Под данным эффектом понимается процесс достижения связанными объектами различной природы общего ритма функций.

Термин синхронизации в условиях рассмотрения связанных периодических колебательных систем с близкими по величине параметрами определяется как наступление колебания на одной и той же частоте. Подобной колебательной системой может выступать осциллятор - система, совершающая колебания, то есть показатели которой периодически повторяются во времени.

Синхронизация связанных осцилляторов является одной из основных тем нелинейной динамики, хорошо изученной для классических периодических систем (например, Ван де осцилляторы Поля), а также для современных хаотических систем (например, осцилляторы Лоренца и Ресслера).

Данная работа посвящена рассмотрению системы, состоящей из мобильных агентов, взаимодействующих с другими агентами, движущихся в трехмерном пространстве. Предмет изучения данной работы – система связанных осцилляторов Ресслера с разными частотами, процесс наступления глобальной синхронизации в системе.

Глава 2

Постановка задачи

В рамках данной работы будет необходимо:

1. Изучить существующие публикации на выбранную тему
2. Освоить необходимую теоретическую базу
3. Описать различные виды осцилляторов прежде чем переходить к осциллятору Ресллера
4. Рассмотреть поведения осциллятора Ресслера при различных параметрах (таких, которые обеспечивают хаотическое движение)
5. Определить параметр связи синхронизации двух осцилляторов Ресслера
6. Смоделировать наступление глобальной синхронизации нескольких агентов, движения которых определяют осцилляторы Ресслера с различными значениями частот
7. Произвести вывод синхронизированных агентов на траекторию, существенно отличающуюся от траектории синхронизации
8. Написать программу для визуализации изученного процесса
9. Проанализировать результаты

Глава 3

Ряд математических моделей, известных в нелинейной динамике.

3.1 Осциллятор Ресслера

Известная парадигматическая хаотическая система, осциллятор Ресслера, задается уравнениями:

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} &= -wy - z, \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= wx + ay, \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= b + z(x - c). \end{cases} \tag{3.1}$$

Поведение осциллятора Ресслера сильно зависит от значений постоянных параметров. Изменение каждого параметра даёт определённый эффект, в результате чего в системе может возникнуть устойчивая неподвижная точка, предельный цикл и тд.

Данная система будет рассмотрена подробно далее.

3.2 Осциллятор Лоренца

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} &= \sigma(y - x), \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= x(\rho - z) - y, \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy - \beta z. \end{cases} \tag{3.2}$$

Стохастическая модель Лоренца, несмотря на внешнюю простоту (три степени свободы: три обыкновенных дифференциальных уравнения, три константы и три начальных условия), не имеет аналитического решения. Трёхмерная система демонстрирует большое разнообразие качественно различных динамических режимов, в том числе сосуществование периодических аттракторов и переход к хаосу через удвоение периода. При $\sigma = 10$, $\beta = 8/3$ и произвольном ρ, система (3.2) является детерминированной.

Обычно исследование системы Лоренца проводят при $\sigma = 10$, $\beta = 8/3$, $\rho = 28$, x(0)=1, y(0)=0, z(0)=0 (классические значения параметров). В этом случае она ведёт себя псевдослучайным (хаотическим) образом.

3.3 Осциллятор Ван дер Поля

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} &= y, \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= \mu(1 - x^2)y - wx, \end{cases} \tag{3.3}$$

w - частота колебаний, μ - параметр нелинейности ≥ 0.

Осциллятор Ван дер Поля является основной моделью для анализа периодических автоколебаний. Это - классическое уравнение теории колебаний, посредством которого описывается универсальный механизм возникновения автоколебательных режимов.

3.4 Характер поведения систем. Графики.

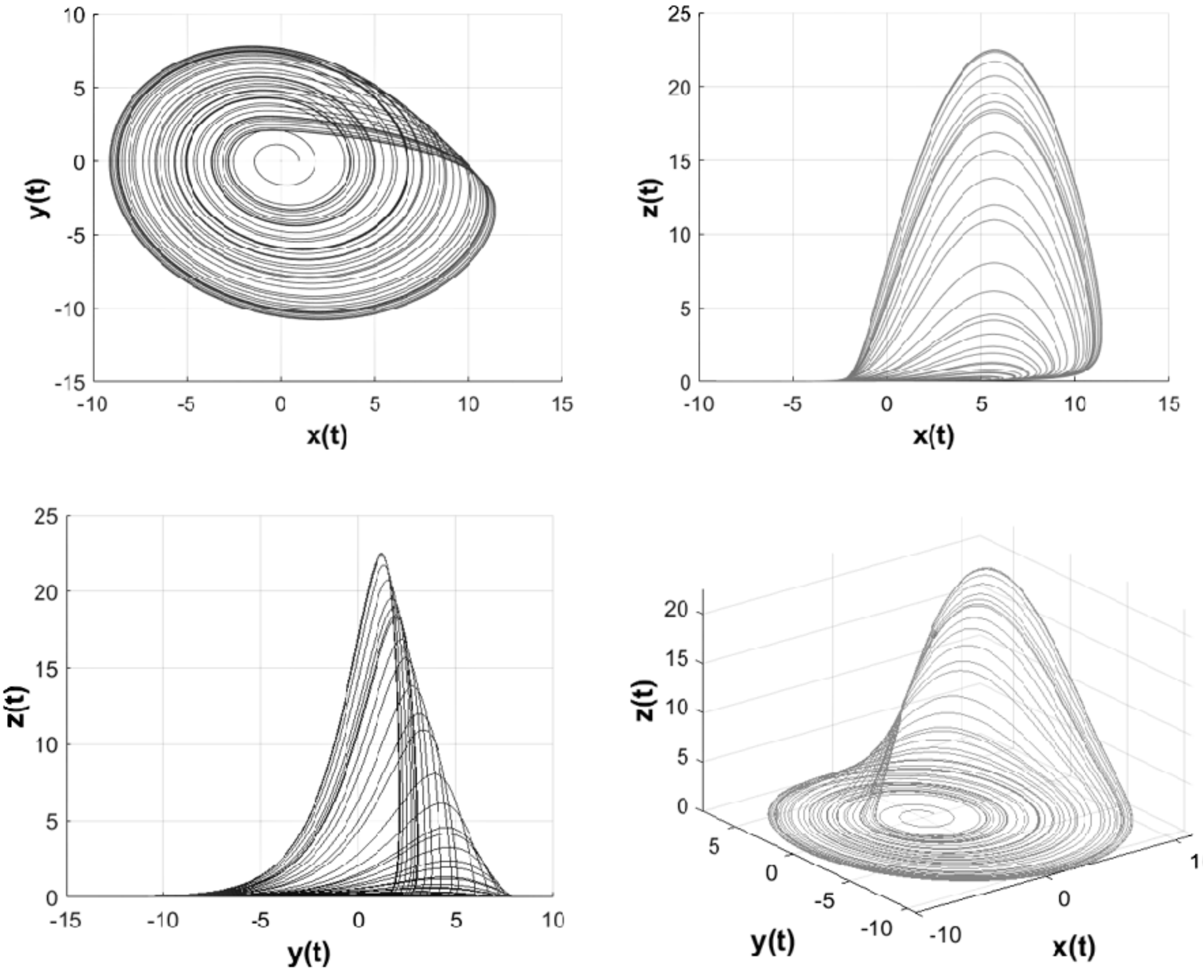


Рисунок 3.1: Характер поведения системы осциллятора Ресслера

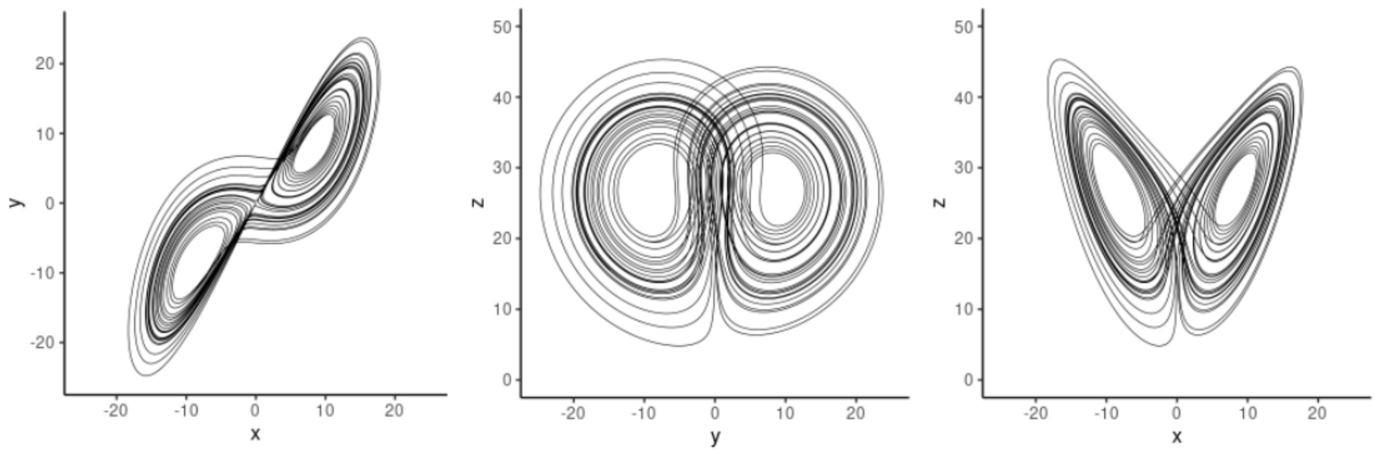


Рисунок 3.2: Характер поведения системы осциллятора Лоренца

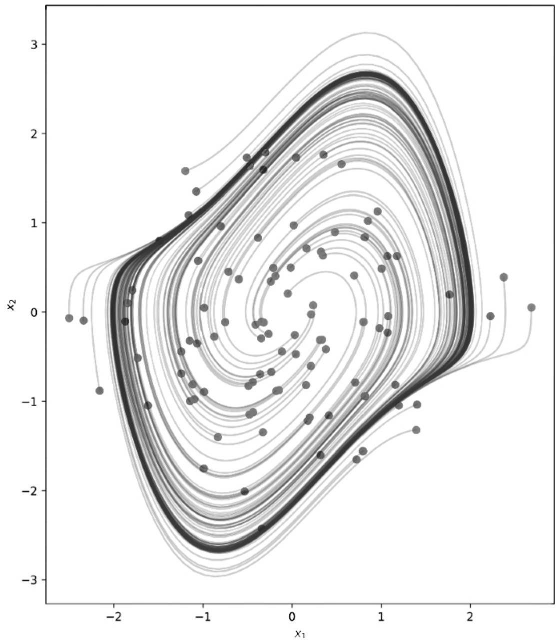


Рисунок 3.3: Характер поведения системы осциллятора Ван дер Поля

Глава 4

Теоретический анализ

xcolor

Рассмотрим осциллятор Ресслера:

$$\begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} &= -wy - z, \\ \frac{\mathrm{d}y}{\mathrm{d}t} &= wx + ay, \\ \frac{\mathrm{d}z}{\mathrm{d}t} &= b + z(x - c). \end{cases} \tag{4.1}$$

где аргументы $w,\ a,\ b,\ c$ - положительные числа.

Для проведения исследования системы возьмем значения коэффициентов $\{\mathrm{w} = 0.98\ ,\ \mathrm{c} = 8.5,\ \mathrm{a} = 0.22,\ \mathrm{b} = 0.1\}$. При этих значениях возникает хаотический аттрактор.

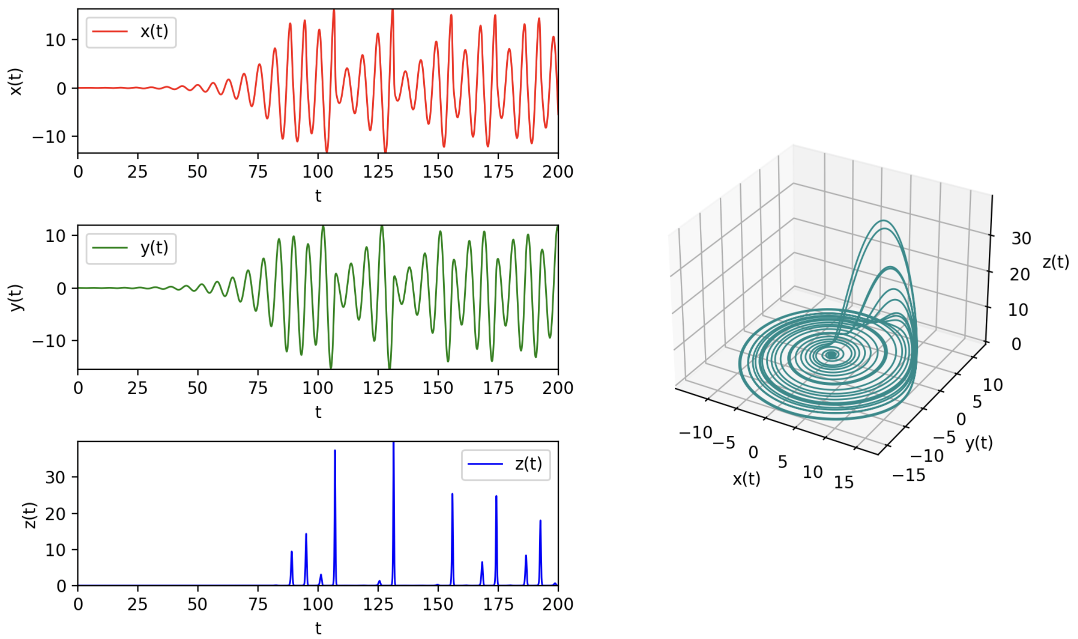


Рисунок 4.1: Осциллятор Ресслера и изменение координат по x, y и z в зависимости от времени.

Будем рассматривать визуализацию, полученную с помощью специально созданной тестовой программы.

Данная работа направлена на изучение процесса синхронизации системы свзязанных осцилляторов Ресслера. То есть, на следующий вариант:

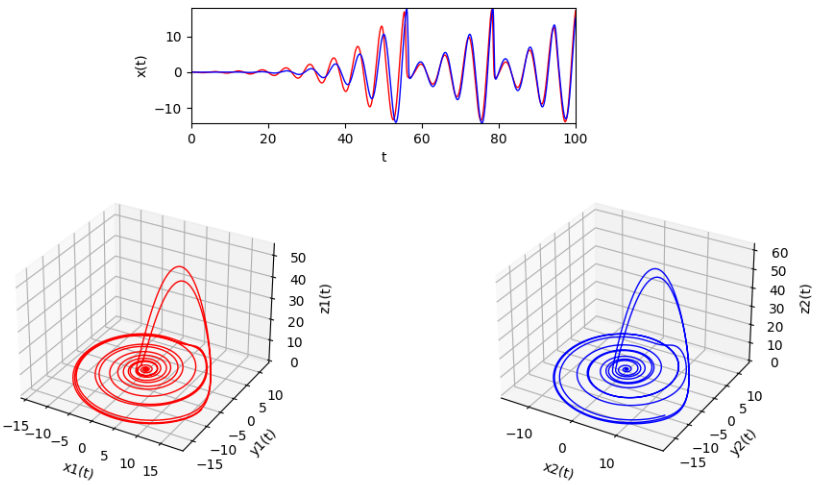


Рисунок 4.2: Два осциллятора, на примере которых будем рассматривать явление синхронизации.

Угол вращения определяет фазу колебаний, которая представляет собой линейно возрастающую функцию (см. Рис 4.3):

$$\phi = \arctan(\frac{\dot{y}}{\dot{x}}); \tag{4.2}$$

Тогда угловая скорость (мгновенная частота) будет вычисляться следующим образом (см. Рис 4.4):

$$v = \frac{\mathrm{d}\phi}{\mathrm{d}t} = \frac{\dot{y}\ddot{x} - \dot{x}\ddot{y}}{\dot{x}^2 + \dot{y}^2} \tag{4.3}$$

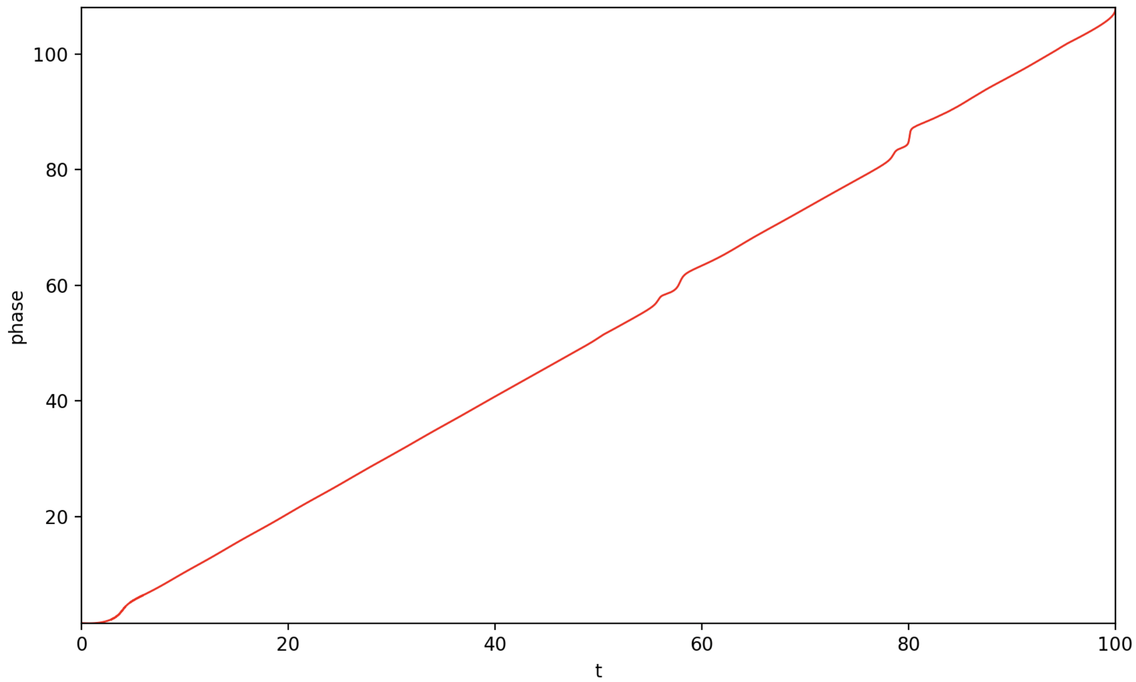


Рисунок 4.3: График зависимости фазы колебаний от времени. Монотонно возрастающая функция.

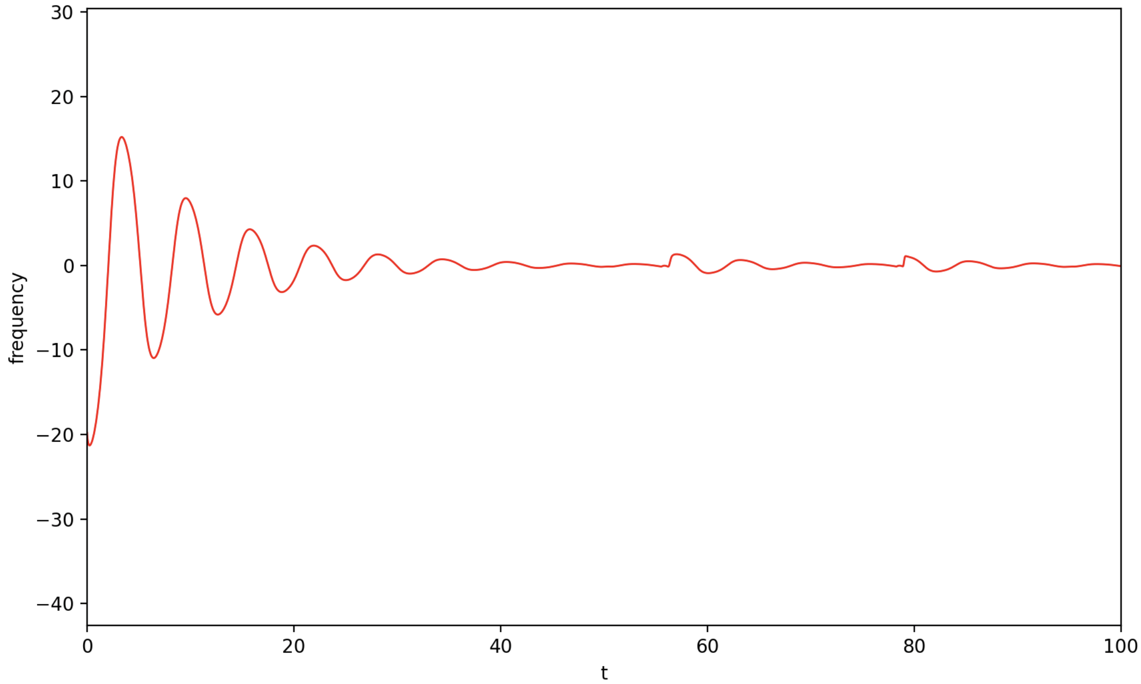


Рисунок 4.4: График зависимости мгновенной частоты от времени.

Изучим систему из двух осцилляторов с близкими по значению параметрами, отличающимися лишь в значении w ($w_1 = 1.02$, $w_2 = 0.98$) - средняя частота осцилляторов. Пусть система задается уравнениями, содержащими новый параметр - коэффициент свзязи d:

$$\begin{cases} \frac{\mathrm{d}x_{1,2}}{\mathrm{d}t} &= -w_{1,2}y_{1,2} - z_{1,2}, \\ \frac{\mathrm{d}y_{1,2}}{\mathrm{d}t} &= w_{1,2}x_{1,2} + ay_{1,2} + d(y_{2,1} - y_{1,2}), \\ \frac{\mathrm{d}z_{1,2}}{\mathrm{d}t} &= b + z_{1,2}(x_{1,2} - c). \end{cases} \tag{4.4}$$

Для иллюстрации перехода к фазовой синхронизации будем наблюдать за изменением частот колебаний (4.5) при увеличении параметра связи d:

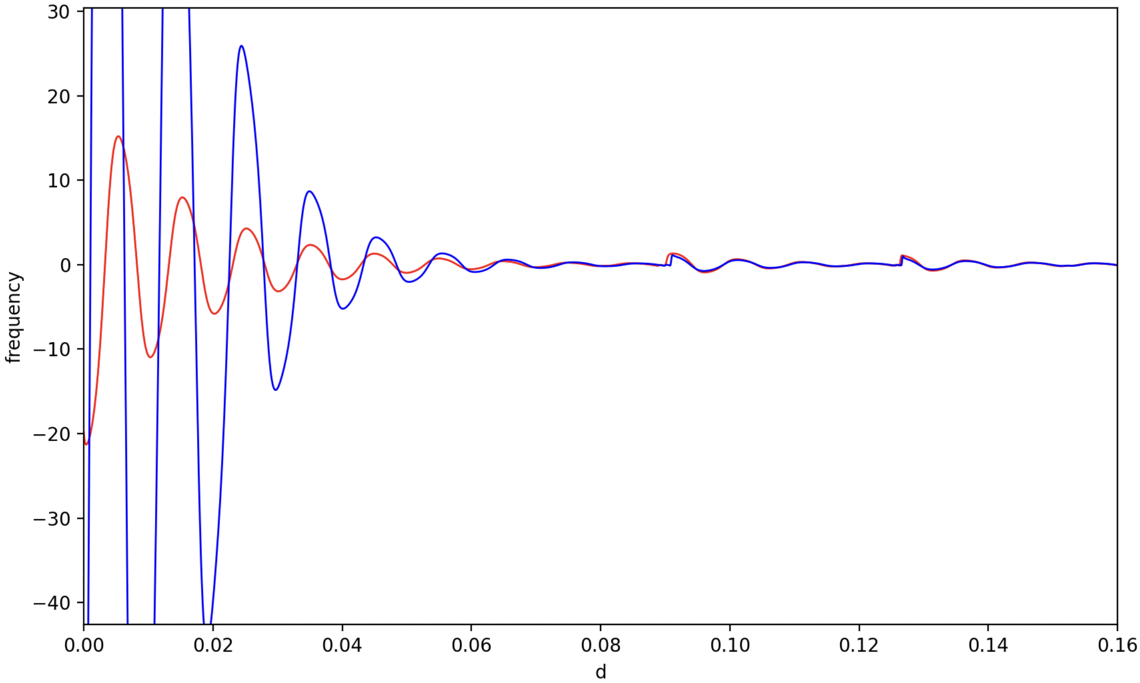


Рисунок 4.5: Синхронизация мгновенных частот. Где d - коэффициент связи осцилляторов.

Легко увидеть, что возникновение фазового захвата происходит при переходе d значения d*, примерно равного 0.06.

Глава 5

Моделирование системы из N взаимосвязанных осцилляторов Ресслера

Изменим систему уравнений (4.4) таким образом, чтобы она удовлетворяла необходимым условиям нашего следующего рассматриваемого случая.

$$\begin{cases} \frac{\mathrm{d}x_i}{\mathrm{d}t} &= -w_i y_i - z_i, \\ \frac{\mathrm{d}y_i}{\mathrm{d}t} &= w_i x_i + a y_i + \Sigma_j d(y_j - y_i), \\ \frac{\mathrm{d}z_i}{\mathrm{d}t} &= b + z_i(x_i - c). \end{cases} \tag{5.1}$$

Наложим границы на значение параметра d, теперь он будет зависеть от расстояния между осцилляторами:

$$d = \begin{cases} \overline{\mathrm{d}} = const, if(\mathrm{x}_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2 < r^2, \\ 0, otherwise. \end{cases} \tag{5.2}$$

Пусть

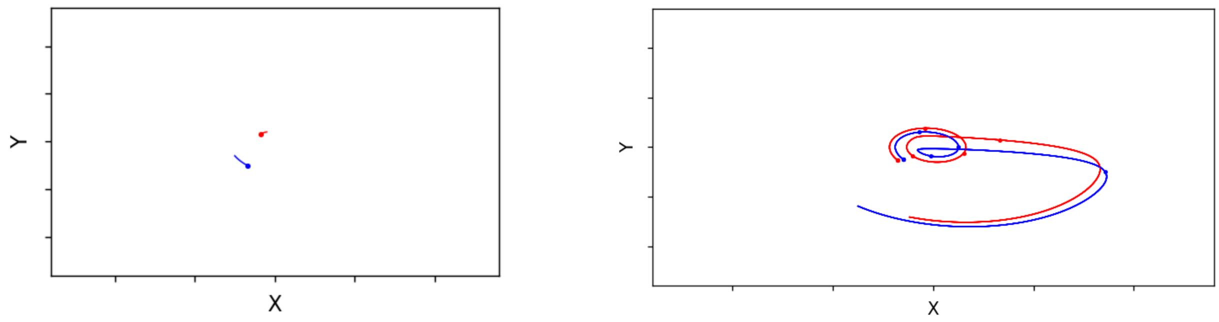
1. $\mathrm{w}_i \in [0,93..1,07]$.

2. В начальный момент времени агенты располагаются в случайных точках.

5.1 Система осцилляторов N = 2

Для начала рассмотрим систему из двух осцилляторов в разные моменты времени, используя (5.1 и 5.2). В начальный момент агенты располагаются в случайных точках, см. Рис. 5.1 (а).

С течением времени некоторые агенты сближаются, тем самым активируя связь. Это приводит к образованию групп агентов, синхронизированных между собой, см. Рис. 5.1 (b) (в случае двух агентов мы получаем одну синхронизированную группу).



(a) (b)

Рисунок 5.1: Расположение агентов на плоскости и след их движения

Увидеть момент синхронизации можно на следующем графике:

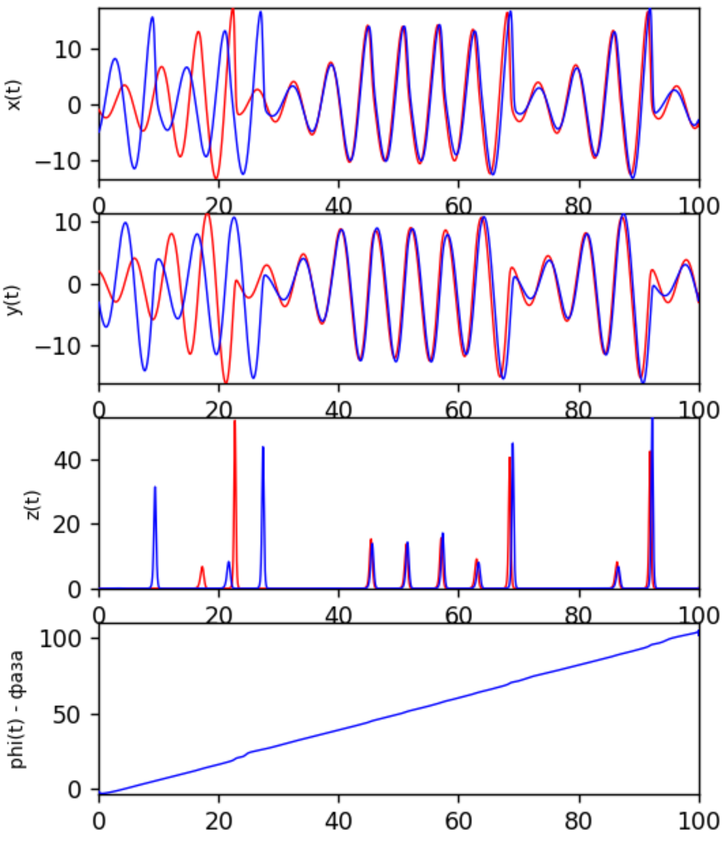


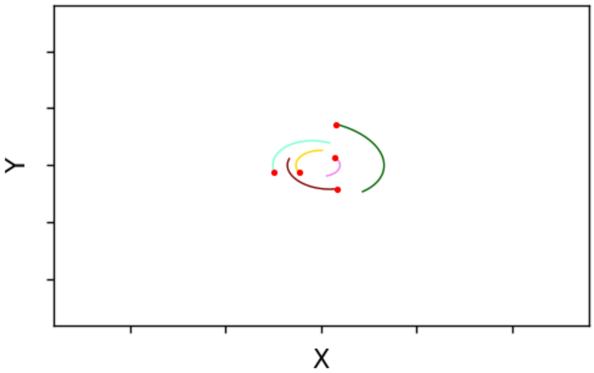
Рисунок 5.2: Изменения x, y, z в зависимости от времени. Наблюдения за процессом синхронизации.

Далее мы рассмотрим более наглядные примеры, включающие в себя N > 2 агентов.

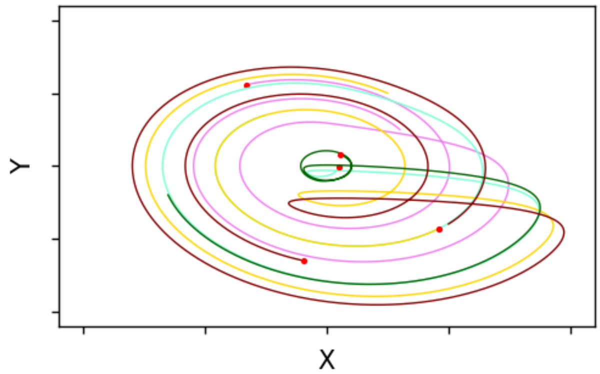
5.2 Система осцилляторов N > 2

Как и в предыдущем случае, с течением времени агенты активируют связь. Это приводит к образованию групп агентов, синхронизированных между собой, см. Рис. 5.3 (b), (c); 5.4 (c) - локальная синхронизация.

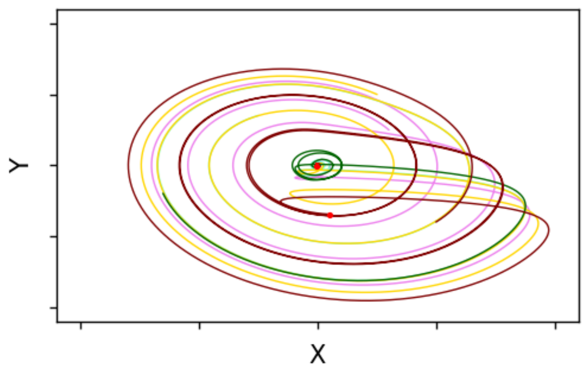
Далее сближаются и синхронизируются уже группы агентов, что в итоге приводит к глобальной синхронизации, см. Рис. 5.3 (e).



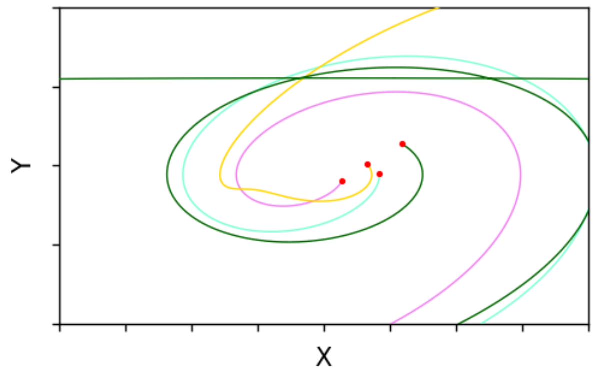
(a)



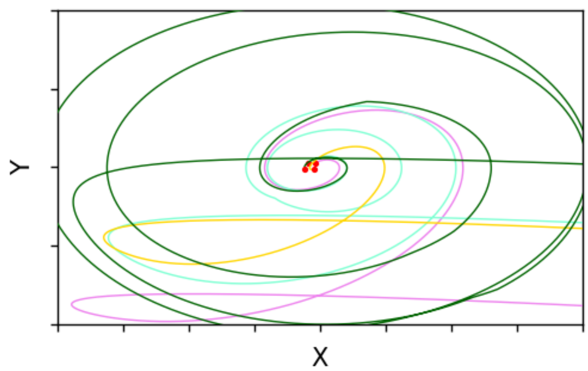
(b)



(c)

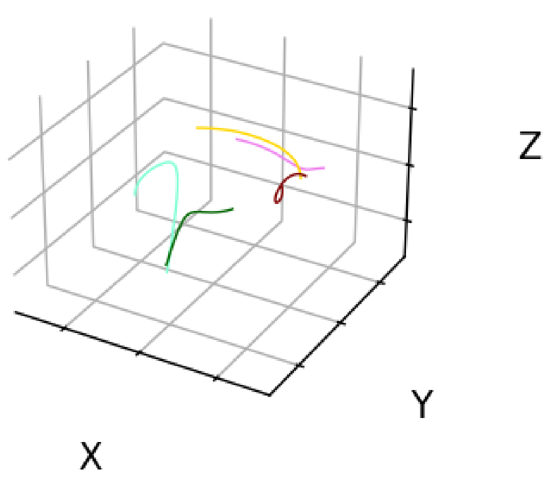


(d)

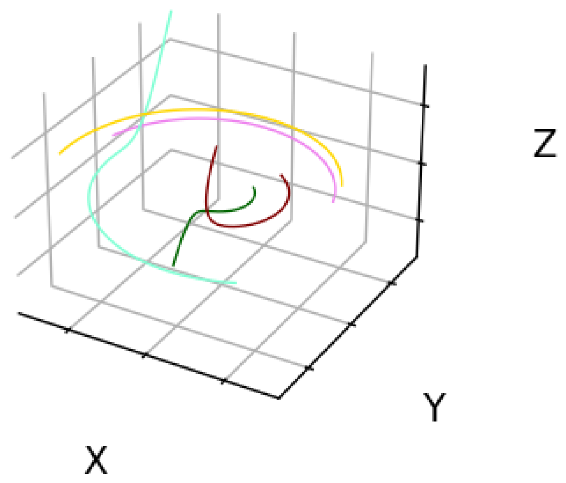


(e)

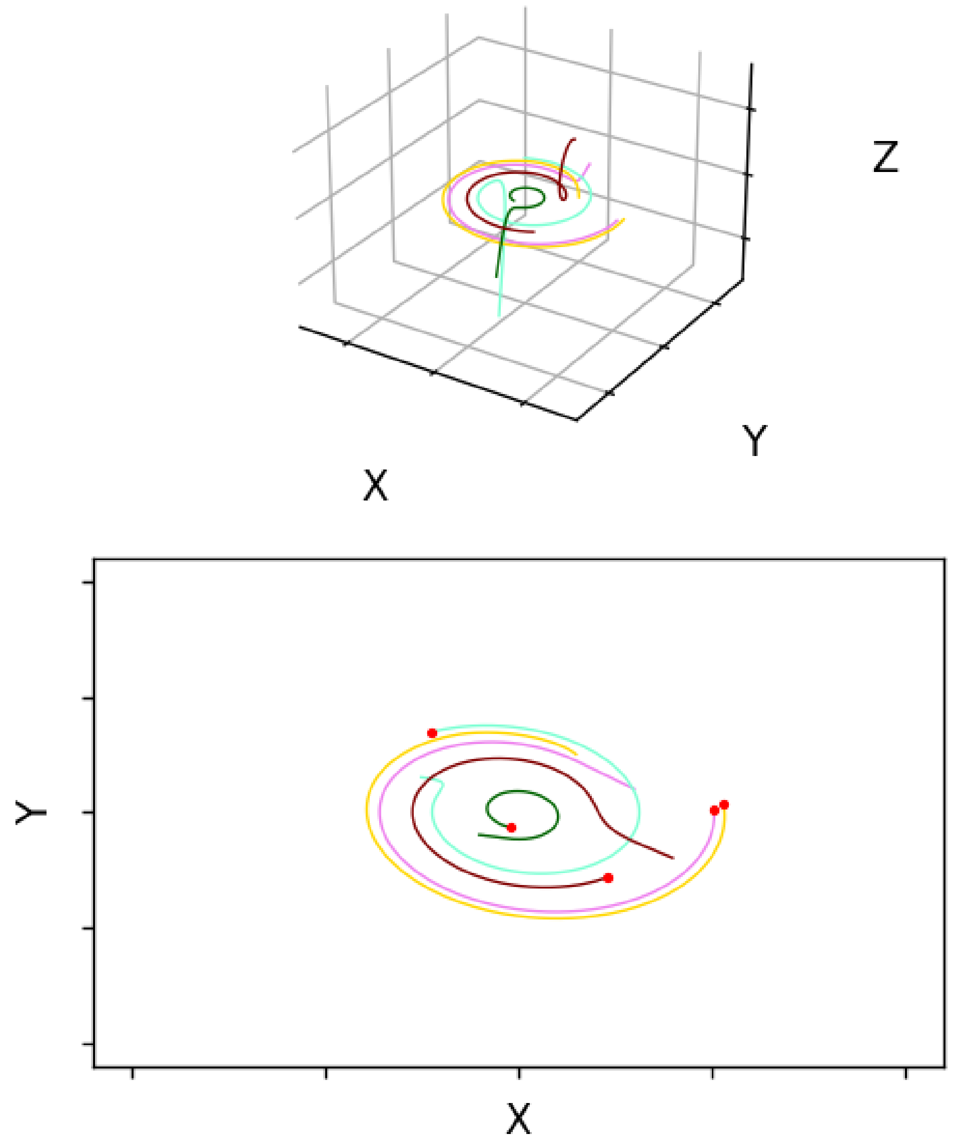
Рисунок 5.3: Расположение агентов на плоскости и след их движения (проекция на плоскость <x, y>). Начальные точки выбраны случайным образом.
w1 = 0.98, w2 = 1, w3 = 0.94, w4 = 1.07, w5 = 1.02



(a)

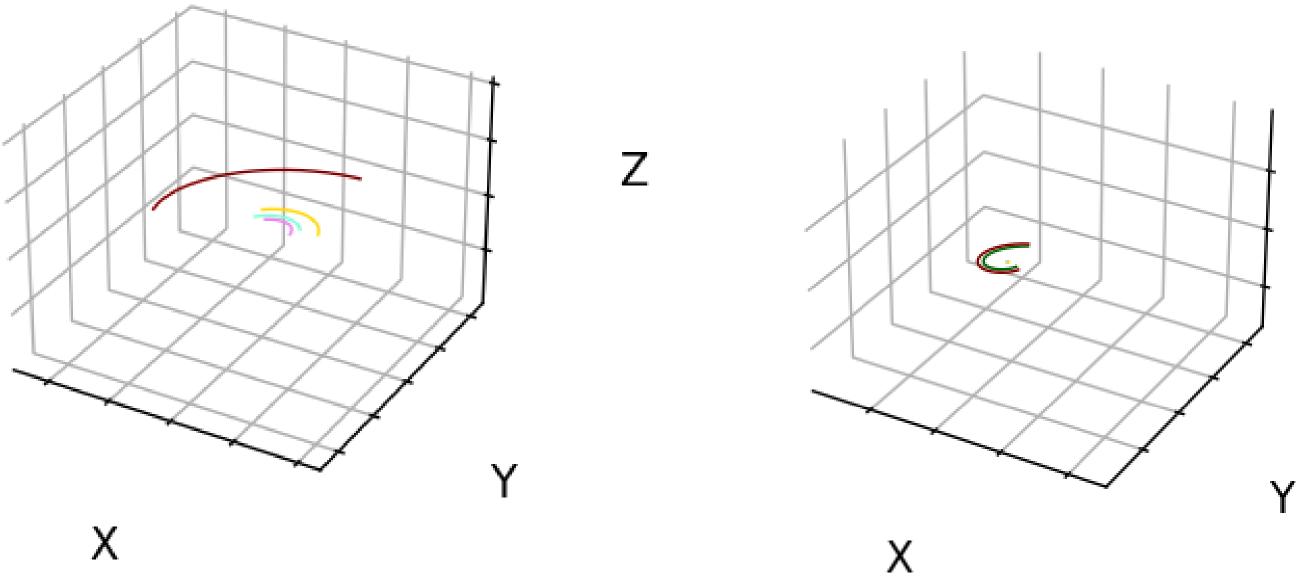


(b)



(c)

Рисунок 5.4: Расположение агентов на плоскости и след их движения (в пространстве).
Начальные точки выбраны случайным образом.
w1 = 0.98, w2 = 1, w3 = 0.94, w4 = 1.07, w5 = 1.02



(a) (b)

Рисунок 5.5: Синхронизация - локальная и глобальная

Увидеть момент синхронизации можно на следующем графике:

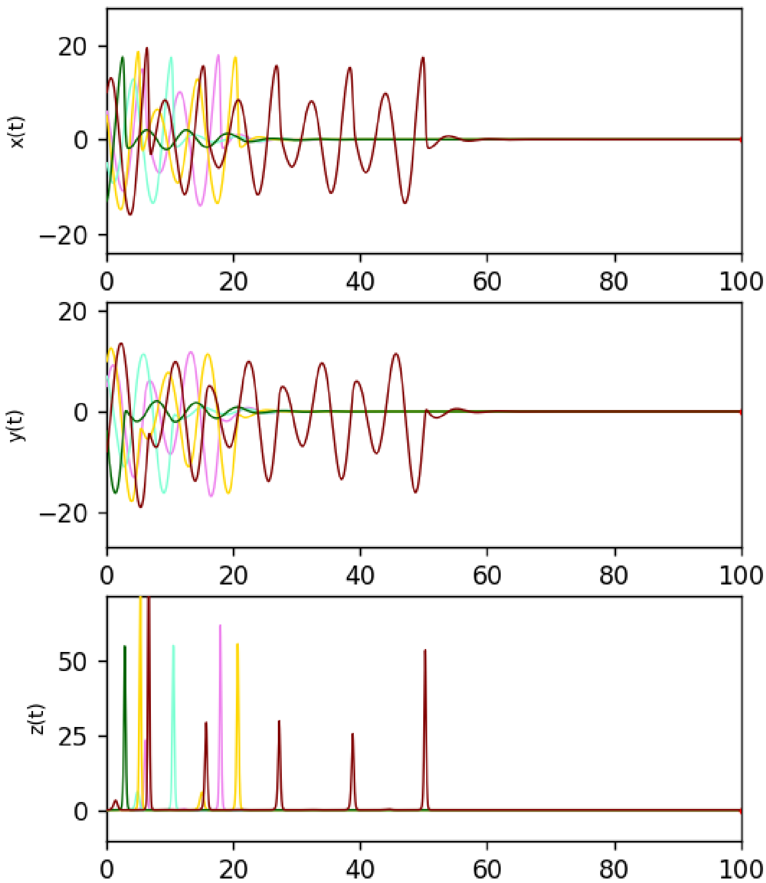
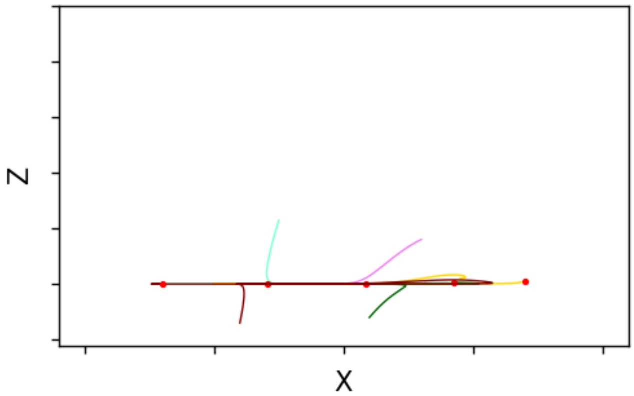
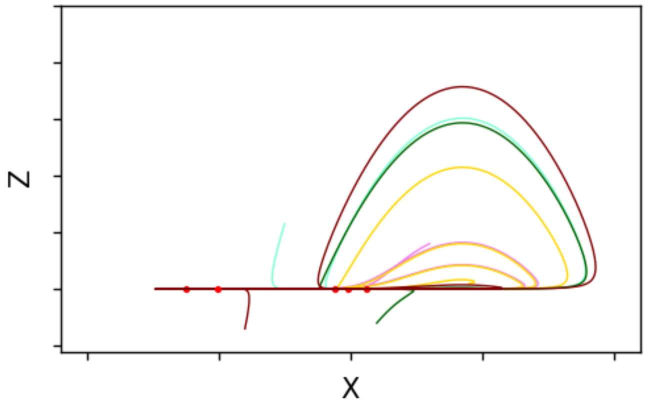


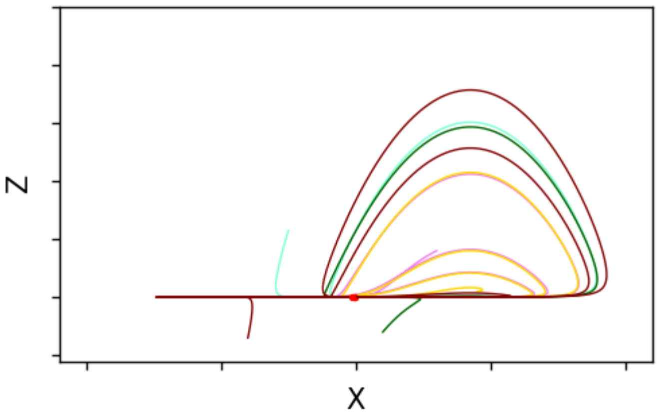
Рисунок 5.6: Изменения x, y, z в зависимости от времени. Наблюдения за процессом синхронизации.



(a)



(b)



(c)

Рисунок 5.7: Расположение агентов на плоскости и след их движения (проекция на плоскость <x, z>). Начальные точки выбраны случайным образом.
w1 = 0.98, w2 = 1, w3 = 0.94, w4 = 1.07, w5 = 1.02

В примере выше мы можем наблюдать такой эффект как осцилляторная смерть. Он возникает в том случае, если связть между элементами цепочки достаточно сильна, тогда их взаимодействие может приводить не только к синхронизации, но и к подавлению автоколебаний. Эффект вымирания автоколебаний тесно связан с эффектом синхронизации.

Глава 6

Организация параллельного движения мобильных агентов

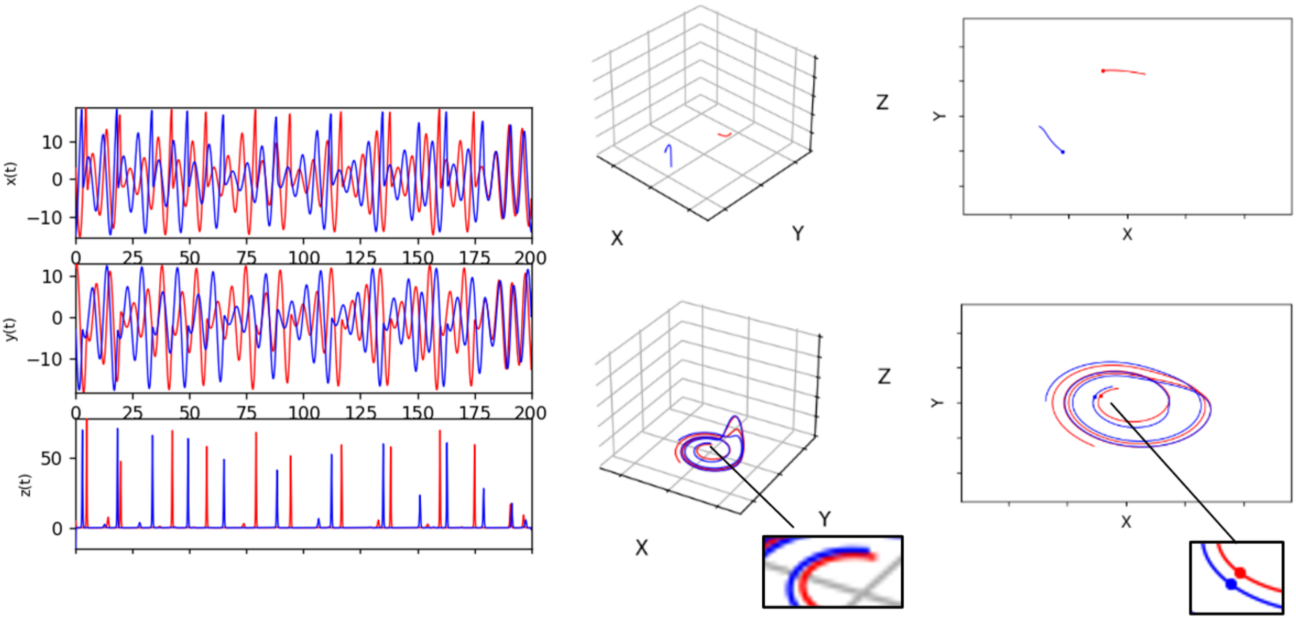
Введем помимо связи синхронизации "отталкивающаю"связь по координате x:

$$\begin{cases} \frac{\mathrm{d}x_i}{\mathrm{d}t} & = -w_i y_i - z_i + \Sigma_j d(x_j - x_i) + \Sigma_j d'/(x_i - x_j), \\ \frac{\mathrm{d}y_i}{\mathrm{d}t} & = w_i x_i + a y_i, \\ \frac{\mathrm{d}z_i}{\mathrm{d}t} & = b + z_i(x_i - c). \end{cases} \tag{6.1}$$

где

$$d, d' = \begin{cases} \overline{\mathrm{d}} = const, if(\mathrm{x}_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2 < r^2, \\ 0, otherwise. \end{cases} \tag{6.2}$$

Эта связь работает следующим образом: при сближении i-го и j-го агентов $|d'/(x_i - x_j)|$ величина растет. Тогда сила взаимодействия, входящая в уравнение i-го агента, < 0 , а соответствующая сила взаимодействия, входящая в уравнение j-го агента, > 0. Таким образом, при сближении пары мобильных агентов предлагаемая связь обеспечит противоположно направленные силы, что приводит агентов к отталкиванию.



(a)

Рисунок 6.1: Расположение агентов на плоскости и след их движения, N = 2. Начальные точки выбраны случайным образом.
w1 = 0.98, w2 = 1, w3 = 0.94, w4 = 1.07, w5 = 1.02

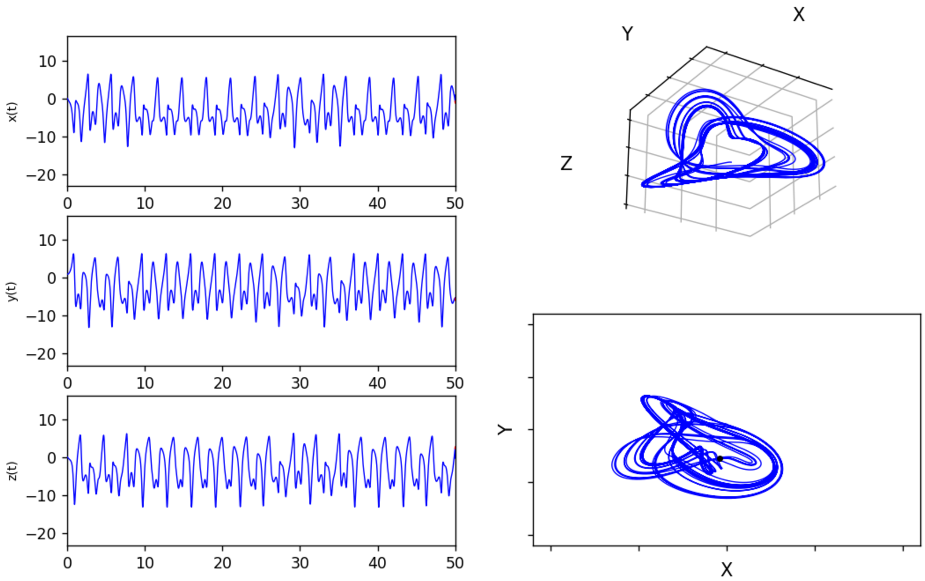
Глава 7

Вывод синхронизированных агентов на траекторию, существенно отличающуюся от траектории синхронизации

В качестве целевой траектории возьмем хаотический Халворсен аттрактор:

$$\begin{cases} \frac{\mathrm{d}X}{\mathrm{d}t} &= -aX - 4Y - 4Z - Y^2, \\ \frac{\mathrm{d}Y}{\mathrm{d}t} &= -aY - 4Z - 4X - Z^2, \\ \frac{\mathrm{d}Z}{\mathrm{d}t} &= -aZ - 4X - 4Y - X^2. \end{cases} \tag{7.1}$$

где a = 1.4.



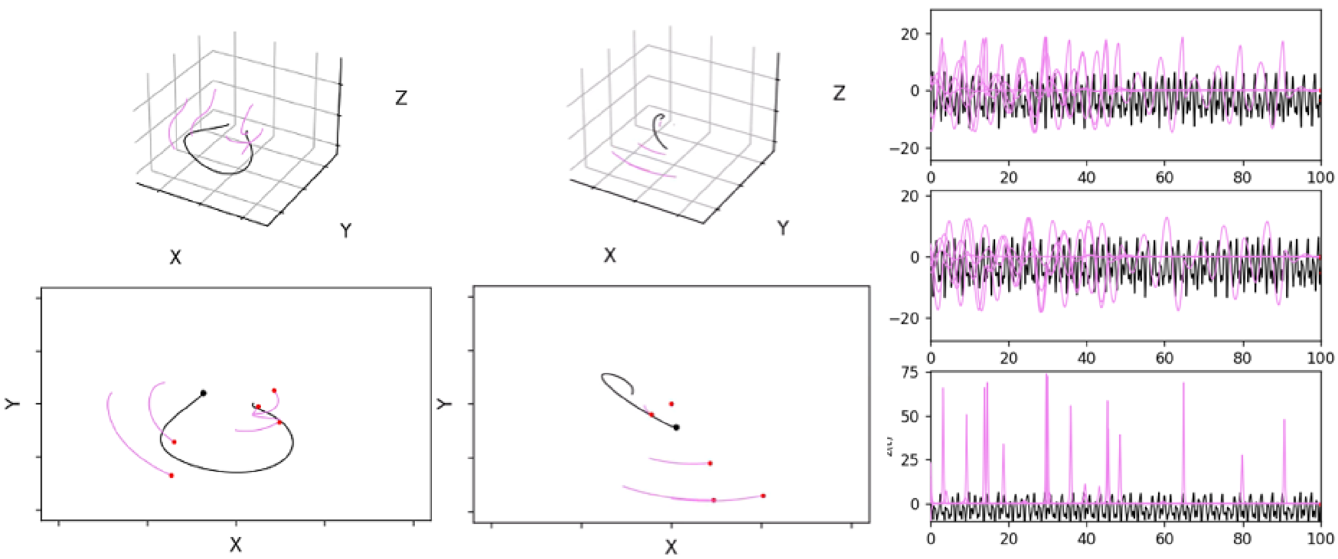
(a)

Рисунок 7.1: Целевая траектория - Халворсен аттрактор

Для достижения синхронизации двух неидентичных систем оказалось необходимо добавить связь по координате x, помимо связи по y:

$$\begin{cases} \frac{\mathrm{d}x_i}{\mathrm{d}t} &= -w_i y_i - z_i + D(X - x_i), \\ \frac{\mathrm{d}y_i}{\mathrm{d}t} &= w_i x_i + a y_i + \Sigma_j d(y_j - y_i) + D(Y - y_i), \\ \frac{\mathrm{d}z_i}{\mathrm{d}t} &= b + z_i(x_i - c) + D(Z - z_i). \end{cases} \tag{7.2}$$

В этом случае, мобильный агент, соответствующий Халворсен аттрактору, постепенно начнет захватывать остальных мобильных агентов, при этом оставаясь на собственной траектории.

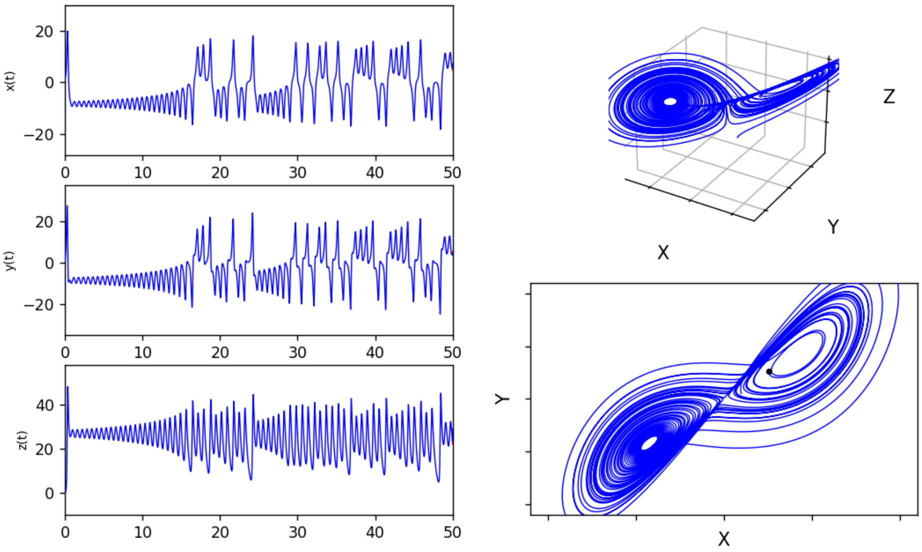


(a)

Рисунок 7.2: Расположение агентов и след их движения в пространстве. N > 2. Аттракторы Ресслера – розовый цвет, целевая траектория – чёрный.

Теперь в качестве целевой траектории возьмем хаотический аттрактор Лоренца:

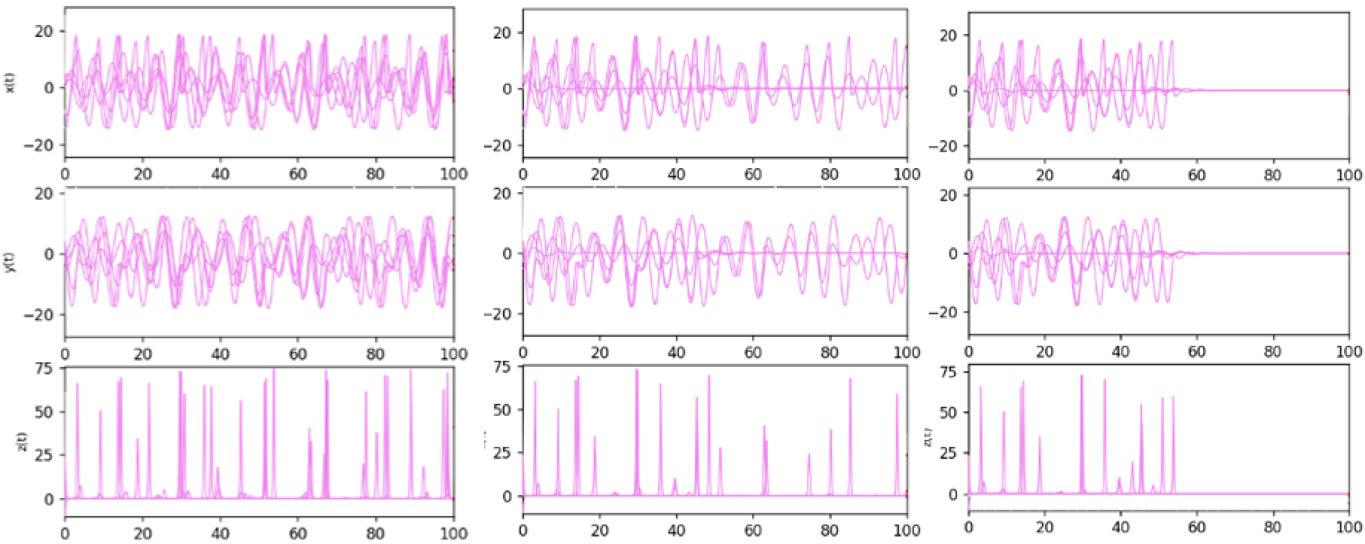
$$\begin{cases} \frac{\mathrm{d}X}{\mathrm{d}t} &= \sigma(Y - X), \\ \frac{\mathrm{d}Y}{\mathrm{d}t} &= X(\rho - Z) - Y, \\ \frac{\mathrm{d}Z}{\mathrm{d}t} &= XY - \beta Z. \end{cases} \tag{7.3}$$



(a)

Рисунок 7.3: Целевая траектория - аттрактор Лоренца

// To be added



(a)

Рисунок 7.4: Слева-направо: нет внешнего влияния; целевая траектория – аттрактор Лоренца; целевая траектория – Халворсен аттрактор

Глава 8

Заключение

Результаты, полученные в ходе выполнения данной работы:

1. Смоделирован процесс глобальной синхронизации ансамбля хаотических осцилляторов
2. Были найдены фаза и частота хаотической системы Ресслера
3. Найдены значения параметра связи, гарантирующие возникновение фазовой синхронизации двух осцилляторов Ресслера с малой частотной расстройкой
4. Введена третья размерность в вычисление коэффициента связи, чего не было в рассмотренных статьях
5. Обнаружен эффект вымирания автоколебаний
6. Было смоделировано управление мобильными агентами, заставляющее их выйти на заданную траекторию движения
7. Найден метод, позволяющий организовать параллельное движение мобильных агентов
8. Было показано, что добавление различных динамических связей как между агентами, так и с «внешним» агентом позволит получить разные структуры, различные траектории. Это позволяет адаптивно управлять агентами.

Глава 9

Реализация практической части

9.1 Исходный код

```
import numpy as np
import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation
from mpl_toolkits.mplot3d import Axes3D
from scipy.signal import hilbert

import math

def num_rossler(x1_n, y1_n, z1_n, h, a, b, c, w1, d_it):
    def d(x1_n, x2_n, y1_n, y2_n):
        d_val = 0
        if ((x1_n - x2_n)**2 + (y1_n - y2_n)**2 < 3**2):
            d_val = 0.2

        return d_val

    # --- connection : d_it = {0.2 or 0}

    dx_1 = ((-w1) * y1_n - z1_n)
    dy_1 = (w1 * x1_n + a * y1_n) # + d(x1_n, x2_n, y1_n, y2_n) * (y2_n - y1_n)
        )
    dz_1 = (b + z1_n * (x1_n - c))

    x1_n1 = x1_n + h * dx_1
    y1_n1 = y1_n + h * dy_1
    z1_n1 = z1_n + h * dz_1

    return x1_n1, y1_n1, z1_n1
```

```python

def num_rossler_freq(x1_n, y1_n, z1_n, h, a, b, c, w1, d_it):
    def d(x1_n, x2_n, y1_n, y2_n):
        d_val = 0
        if ((x1_n - x2_n)**2 + (y1_n - y2_n)**2 < 3**2):
            d_val = 0.2
        # d_val = d_it

        return d_val

    # --- connection : d_it = {0.2 or 0}

    dx_1 = ((-w1) * y1_n - z1_n)
    dy_1 = (w1 * x1_n + a * y1_n) # + d(x1_n, x2_n, y1_n, y2_n) * (y2_n - y1_n)
        )
    dz_1 = (b + z1_n * (x1_n - c))

    dx_1_v = ((-w1) * y1_n - z1_n)
    dy_1_v = (w1 * x1_n + a * y1_n)
    dz_1_v = (b + z1_n * (x1_n - c))

    x1_n1 = x1_n + h * dx_1
    y1_n1 = y1_n + h * dy_1
    z1_n1 = z1_n + h * dz_1

    phi_1 = np.arctan(dy_1/dx_1)
    phi_1 = np.arctan(y1_n/x1_n)

    dphi_1 = ((w1 + a) * dx_1 + (-w1 - 1) * dy_1) / (dx_1**2)

    dphi_f_1 = 1 / (1 + (dy_1/dx_1)**2)
    dphi_g_1 = dphi_1
    res_dphi1 = dphi_f_1 * dphi_g_1

    d_dx = (-1)*w1 - 1
    d_dy = w1 + a

    first = dy_1_v * d_dx - dx_1_v * d_dy
    sec = dx_1_v**2 + dy_1_v**2

    res_dphi1 = (first/sec)

    phase_diff_1 = phi_1

```

```
    return res_dphi1, phase_diff_1


def rossler_eq(x, y, z, a, b, c, w):
    #
    dx = (-w) * y - z
    dy = w * x + a * y
    dz = b + z * (x - c)

    phi_1 = np.arctan(dy/dx)

    d_dx = (-1)*w - 1
    d_dy = w + a

    first = dy * d_dx - dx * d_dy
    sec = dx**2 + dy**2

    res_dphi1 = first / sec

    return dx, dy, dz, res_dphi1


def solve(x, y, z, phi, v, t, a, b, c, w):
    for i in range(len(t) - 1):
        dx, dy, dz, dphi = rossler_eq(x[i], y[i], z[i], a, b, c, w)

        dt = t[i + 1] - t[i]

        phi[i] = np.arctan(dy / dx)
        v[i] = dphi

        x[i + 1] = x[i] + (dx * dt)
        y[i + 1] = y[i] + (dy * dt)
        z[i + 1] = z[i] + (dz * dt)

def visualize(x1, y1, z1, t, freq_1, time_frame, steps_t, steps_fr, N):
    fig = plt.figure(figsize=(10, 6))

    h1 = hilbert(x1[0])
    phi1 = np.unwrap(np.angle(h1))

    col = (np.random.random(), np.random.random(), np.random.random())

    axis_1 = fig.add_subplot(2, 2, 2, projection='3d')
```

```
    line1, = axis_1.plot([], [], [], color=col, linewidth=0.8) # osc

    axis_1_track = fig.add_subplot(2, 2, 4)
    line1_track, = axis_1_track.plot([], [], color=col, linewidth=0.8) # osc

    point_track1, = axis_1_track.plot([], [], marker='o', color=col, markersize=2)

    point, = axis_1.plot([], [], [], marker='o', color=col, markersize=1)

    axis_1.xaxis.set_pane_color((1.0, 0.0, 1.0, 0.0))
    axis_1.yaxis.set_pane_color((1.0, 0.0, 1.0, 0.0))
    axis_1.zaxis.set_pane_color((1.0, 0.0, 1.0, 0.0))

    axis_1.set_xlim(-28, 28)
    axis_1.set_ylim(-28, 28)
    axis_1.set_zlim(0, 50)

    axis_1_track.set_xlim(-28, 28)
    axis_1_track.set_ylim(-28, 28)

    axis_1.set_xlabel('X', fontsize=12)
    axis_1.set_ylabel('Y', fontsize=12)
    axis_1.set_zlabel('Z', fontsize=12)

    axis_1_track.set_xlabel('X', fontsize=12)
    axis_1_track.set_ylabel('Y', fontsize=12)

    axis_1.set_yticklabels([])
    axis_1.set_xticklabels([])
    axis_1.set_zticklabels([])

    axis_1_track.set_yticklabels([])
    axis_1_track.set_xticklabels([])

    axis_2_x = fig.add_subplot(4, 2, 1)
    line3_1_x, = axis_2_x.plot([], [], 'b-', linewidth=0.8) # osc 1

    point3_1_x, = axis_2_x.plot([], [], marker='o', color=col, markersize=1)

    axis_2_y = fig.add_subplot(4, 2, 3)
    line3_1_y, = axis_2_y.plot([], [], 'b-', linewidth=0.8) # osc 1

    point3_1_y, = axis_2_y.plot([], [], marker='o', color=col, markersize=1)
```

```
axis_2_z = fig.add_subplot(4, 2, 5)
line3_1_z, = axis_2_z.plot([], [], 'b-', linewidth=0.8) # osc 1

point3_1_z, = axis_2_z.plot([], [], marker='o', color=col, markersize=1)

axis_3 = fig.add_subplot(4, 2, 7)
line_phi, = axis_3.plot([], [], 'g-', linewidth=0.8)

axis_3.set_xlim(0, time_frame)
axis_3.set_ylim(min(phi1), max(phi1))
axis_3.set_ylabel('phi(t) -          ', fontsize=8)
point_phi, = axis_3.plot([], [], marker='o', color='b', markersize=1)

# --- setting the data

axis_2_x.set_xlim(t[0], t[len(t) - 1])
axis_2_x.set_xlim(0, time_frame)
axis_2_x.set_ylim(min(x1[0]) - 10, max(x1[0] + 10))
axis_2_x.set_ylabel('x(t)', fontsize=8)

axis_2_y.set_xlim(t[0], t[len(t) - 1])
axis_2_y.set_xlim(0, time_frame)
axis_2_y.set_ylim(min(y1[0]) - 10, max(y1[0]) + 10)
axis_2_y.set_ylabel('y(t)', fontsize=8)

axis_2_z.set_xlim(t[0], t[len(t) - 1])
axis_2_z.set_xlim(0, time_frame)
axis_2_z.set_ylim(min(z1[0]) - 10, max(z1[0]) + 10)
axis_2_z.set_ylabel('z(t)', fontsize=8)

# when_to_track = round((steps_t/10)*9)
# when_to_track = 0

# x1_val_track = x1[when_to_track:steps_t]
# y1_val_track = y1[when_to_track:steps_t]

colors = ['violet',
          'aquamarine',
          'gold',
          'darkgreen',
          'maroon']

def update_all(i):
```

```
        if (i < steps_t):
            x1_val = x1[n][0:i]
            y1_val = y1[n][0:i]
            z1_val = z1[n][0:i]

            tq = t[0:i]

            v1_val = freq_1[n][0:i]

            line1.set_data(x1_val, y1_val)
            line1.set_3d_properties(z1_val)
            line1.set_3d_properties(z1_val)

            point.set_3d_properties(z1[n][i])
            line1.set_color(colors[n])

            line1_track.set_color(colors[n])

            #------ x pro

            line3_1_x.set_data(tq, x1_val)
            point3_1_x.set_data(t[i], x1[n][i])
            line3_1_x.set_color(colors[n])

            #------ y pro

            line3_1_y.set_data(tq, y1_val)
            point3_1_y.set_data(t[i], y1[n][i])
            line3_1_y.set_color(colors[n])

            #------ z pro

            line3_1_z.set_data(tq, z1_val)
            point3_1_z.set_data(t[i], z1[n][i])
            line3_1_z.set_color(colors[n])

        # --- from the when_to_track point
            # if (i > when_to_track):
                # line1_track.set_data(x1_val_track, y1_val_track)
                # line2_track.set_data(x2_val_track, y2_val_track)
                # line1_track.set_data(x1_val, y1_val)
                # line2_track.set_data(x2_val, y2_val)

                # point_track1.set_data(x1[i], y1[i])
```

```python
                # point_track2.set_data(x2[i], y2[i])

        # --- from the start
            line1_track.set_data(x1_val, y1_val)

            point_track1.set_data(x1[n][i], y1[n][i])
        return line1, line1_track, line3_1_x, line3_1_y, line3_1_z, point, point_track1, point3_1_x, point3_1_y, point3_1_z

    def update_phi(j):
        j = (j - 1) * 100 - 1

        if (j < steps_t & j < steps_fr):
            tq = t[0:j]

            phi_val = phi1[0:j]

            line_phi.set_data(tq, phi_val)
            point_phi.set_data(t[j], phi1[j])
            line_phi.set_color("blue")

        return line_phi, point_phi

    anim_phi = FuncAnimation(fig, update_phi, frames = np.size(x1[0]), interval = 0, blit =True)
    anim = FuncAnimation(fig, update_all, frames = np.size(x1[0]), interval = 0, blit = True)

    plt.show()
    plt.savefig('rosslerAttractor.png')


def main():
    N_of_agents = 5

    a = 0.22
    b = 0.1
    c = 8.5

    t_ini = 0
    t_fin = 100
    h = 0.0001
    numsteps = int((t_fin - t_ini) / h)
    t = np.linspace(t_ini, t_fin, numsteps)
```

```python

    d_ini = 0
    d_fin = 0.16
    d_h = 0.00000016
    d_numsteps = int((d_fin - d_ini) / d_h)
    d = np.linspace(d_ini, d_fin, d_numsteps)

    x1 = np.random.rand(N_of_agents, numsteps)
    y1 = np.random.rand(N_of_agents, numsteps)
    z1 = np.random.rand(N_of_agents, numsteps)

    freq_1 = np.random.rand(N_of_agents, numsteps)
    phase_diff_1 = np.random.rand(N_of_agents, numsteps)

    for i in range(N_of_agents):
        w1=np.random.uniform(0.93,1.07) # random in the range of (0,93 .. 1,07)

        x1[i] = np.zeros(numsteps)
        y1[i] = np.zeros(numsteps)
        z1[i] = np.zeros(numsteps)

        freq_1[i] = np.zeros(numsteps)
        phase_diff_1[i] = np.zeros(numsteps)

        # random start point
        x1[i][0] = np.random.uniform(-5, 5)
        y1[i][0] = np.random.uniform(-5, 5)
        z1[i][0] = np.random.uniform(-5, 5)

        for k in range(numsteps-1):
            [x1[i][k+1], y1[i][k+1], z1[i][k+1]] = num_rossler(x1[i][k], y1[i][
               k], z1[i][k], t[k+1] - t[k], a, b, c, w1, d[k])

        for k in range(d_numsteps-1):
            [freq_1[i][k+1], phase_diff_1[i][k+1]] = num_rossler_freq(x1[i][k],
                y1[i][k], z1[i][k], t[k+1]-t[k], a, b, c, w1, d[k])

    visualize(x1, y1, z1, t, freq_1, t_fin, numsteps, d_numsteps, N_of_agents)


if __name__ == "__main__":
    main()
```

Листинг 9.1: two.py

Глава 10

Список литературы

1. Chaotic Phase Synchronization and Desynchronization in an Oscillator Network for Object Selection. Fabricio A. Breve, Liang Zhao, Marcos G. Quiles, Elbert E. N. Macau

2. Особенности картины синхронизации импульсами в системе с трехмерным фазовым пространством на примере системыРесслер. А.П. Кузнецов, Н.В. Станкевич, Л.В. Тюрюкина

3. Phase Synchronization of Coupled Rossler Oscillators: Amplitude Effect. LI Xiao-Wen and ZHENG Zhi-Gang Department of Physics and the Beijing-Hong-Kong-Singapore Joint Center for Nonlinear and Complex Systems (Beijing), Beijing Normal University, Beijing 100875, Chinфб 2006

4. Piecewise affine models of chaotic attractors: The Rössler and Lorenz systems. Gleison F. V. Amaral, Christophe Letellier, Luis Antonio Aguirrea, 2005

5. Phase Synchronization in Driven and Coupled Chaotic Oscillators. Michael G. Rosenblum, Arkady S. Pikovsky, J. Kurths

6. Controlling the Motion of a Group of Mobile Agents. V. A. Levin and G. V. Osipov, 2015

7. Synchronization and Coherence of Dynamical Systems: Networks of Coupled R¨ossler Attractors. Kyle J. Pounder, Advisor: Dr. Timothy D. Sauer

8. Phase Synchronization In Three-dimensional Lattices And Globally Coupled Populations Of Nonidentical Rossler Oscillators. Limin Qi B.S. Xian Institute of Science and Technology, 1982

9. Phase Synchronization in Small-world Network Composed of Fractional-order Chaotic Oscillator. Feng Chen, Xiaodan Shao and Long Sheng

10. Lindsey W.C. Synchronization systems in communication and control. Englewood Clis,NJ: Prentice-Hall,1972.

11. Sparrow C. The Lorenz Equations: Bifurcations, Chaos, and Strange Attractors. Berlin: Springer-Verlag, 1982.

12. Manneville P., Pomeau Y. Intermittency and Lorenz model // Phys. Lett. A. 1979

13. Ott E. Chaos in dynamical systems. Cambridge: Cambridge University Press, 1992

14. Appleton E.V. The automatic synchronization of triode oscillator // Proc. Cambridge Phil. Soc. (Math and Phys. Sci.) 1922

15. Pikovsky A.S., Rosenblum M.G., Kurths J., Phase synchronization in regular and chaotic systems // Int. J. of Bifurcation and Chaos. 2000

16. Storti D.W., Rand R.H. Dynamics of two strongly coupled Van der Pol oscillators // Int. J. Non-Linear Mech. 1982

17. Rand R.H., Holmes P.J. Bifurcation of periodic motions in two weakly coupled Van der Pol oscillators

18. Hoppensteadt F.C., Izhikevich E.M. Weakly Connected Neural Networks. New York: Springer-Verlag, 1997

19. Osipov G.V., Kurths J. Regular and chaotic phase synchronization of coupled circle maps

20. Osipov G.V., Pikovsky A.S., Kurths J. Phase synchronization of chaotic rotators

21. Bar-Eli K. On the stability of coupled chemical oscillators

22. Belykh V.N., Mosekilde E. One-dimensional map lattices: Synchronization, bifurcations, and chaotic structures

23. Hu B., Liu Z. Phase synchronization of two-dimensional lattices of coupled chaotic maps

24. Ivanchenko M.V., Osipov G.V., Shalfeev V.D., Kurths J. Phase synchronization in ensemble of bursting oscillators