

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»

Выпускная квалификационная работа бакалавра

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

1	Введение	2
2	Постановка задачи	4
3	Ряд математических моделей, известных в нелинейной динамике.	5
3.1	Осциллятор Ресслера	5
3.2	Осциллятор Лоренца	5
3.3	Осциллятор Ван дер Поля	6
3.4	Характер поведения систем. Графики.	7
4	Теоретический анализ	9
5	Моделирование системы из N взаимосвязанных осцилляторов Ресслера	13
5.1	Система осцилляторов $N = 2$	13
5.2	Система осцилляторов $N > 2$	15
6	Организация параллельного движения мобильных агентов	19
7	Вывод синхронизированных агентов на траекторию, существенно отличающуюся от траектории синхронизации	21
8	Заключение	24
9	Реализация практической части	25
9.1	Исходный код	25
10	Список литературы	33

Глава 1

Введение

Открытие сложного непредсказуемого поведения траекторий в системах - динамического хаоса, по праву считается одним из самых значимых достижений науки, благодаря 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, таких как биология, химия, физика, техника и т.д.

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

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, вза-

и 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 осцилятору Ресслера
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:

$$\begin{cases} \frac{dx}{dt} = -wy - z, \\ \frac{dy}{dt} = wx + ay, \\ \frac{dz}{dt} = b + z(x - c). \end{cases} \quad (3.1)$$

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

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x), \\ \frac{dy}{dt} = x(\rho - z) - y, \\ \frac{dz}{dt} = xy - \beta z. \end{cases} \quad (3.2)$$

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

$$\begin{cases} \frac{dx}{dt} = y, \\ \frac{dy}{dt} = \mu(1 - x^2)y - wx, \end{cases} \quad (3.3)$$

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.

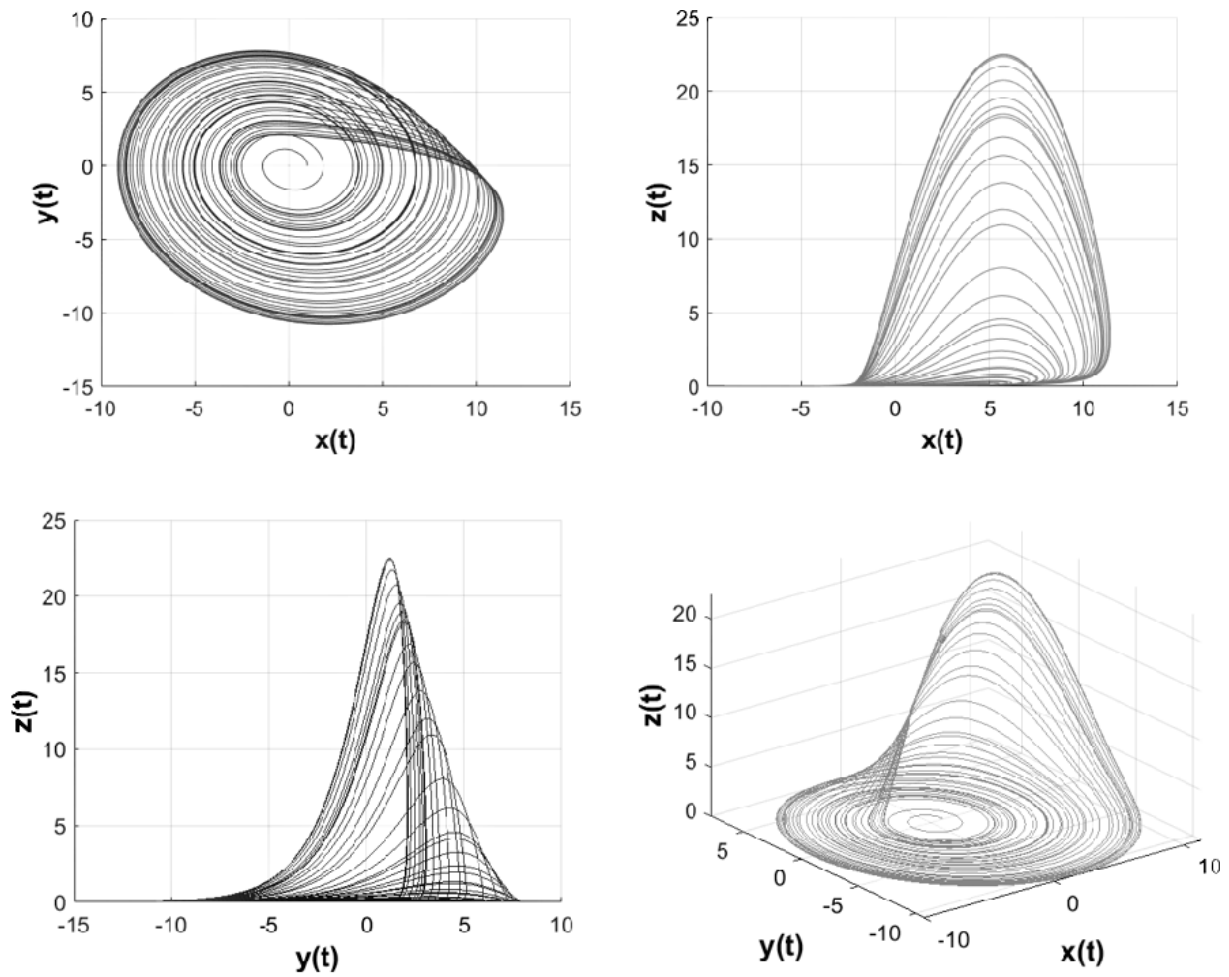


Рисунок 3.1: Характер поведения системы осциллятора Ресслера

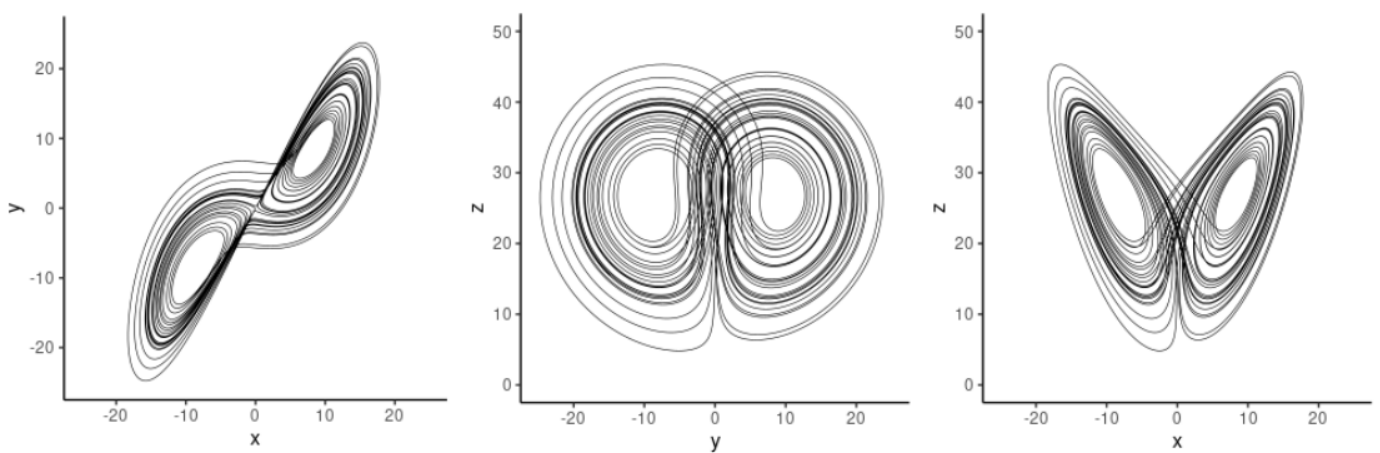


Рисунок 3.2: Характер поведения системы осциллятора Лоренца

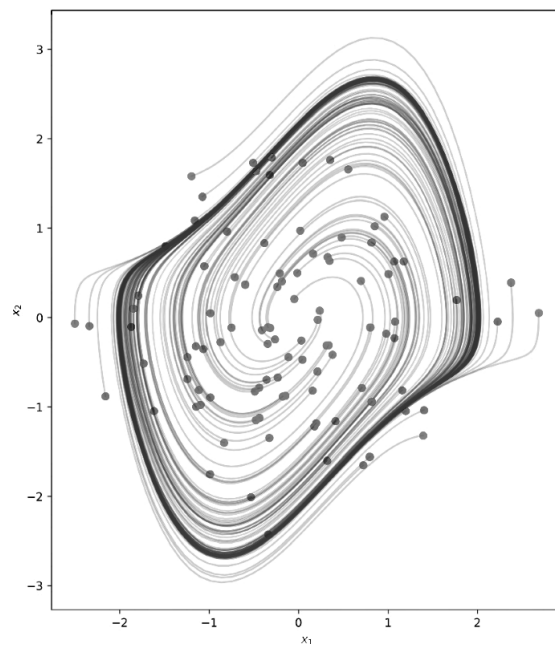


Рисунок 3.3: Характер поведения системы осциллятора Ван дер Поля

Глава 4

Теоретический анализ

хcolor

Рассмотрим осциллятор Ресслера:

$$\begin{cases} \frac{dx}{dt} = -wy - z, \\ \frac{dy}{dt} = wx + ay, \\ \frac{dz}{dt} = b + z(x - c). \end{cases} \quad (4.1)$$

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

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

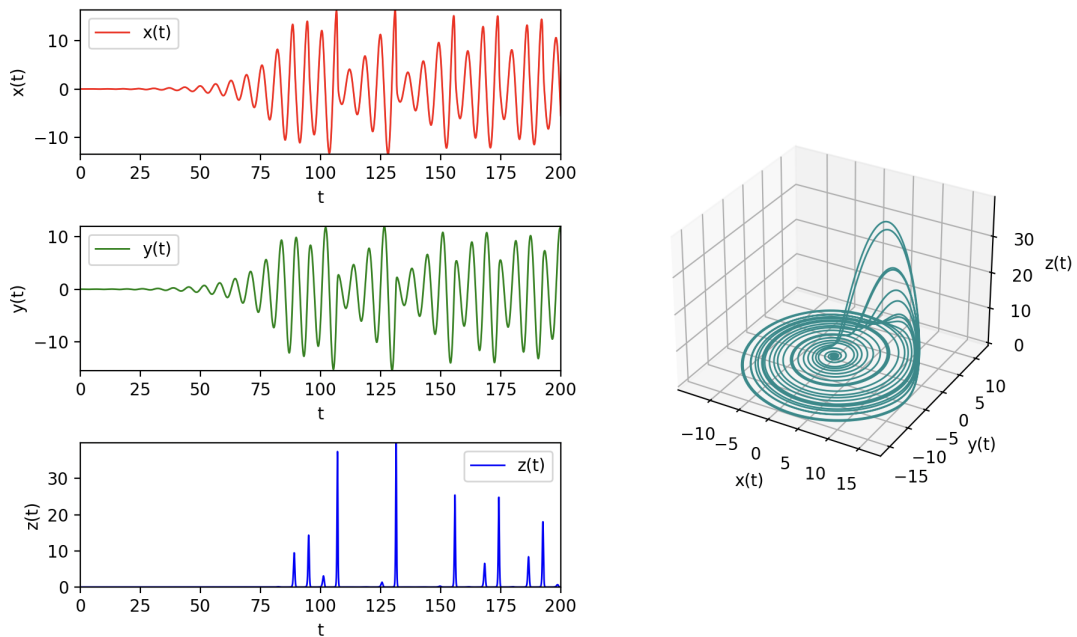


Рисунок 4.1: Осциллятор Ресслера и изменение координат по x , y и z в зависимости от времени.

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

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

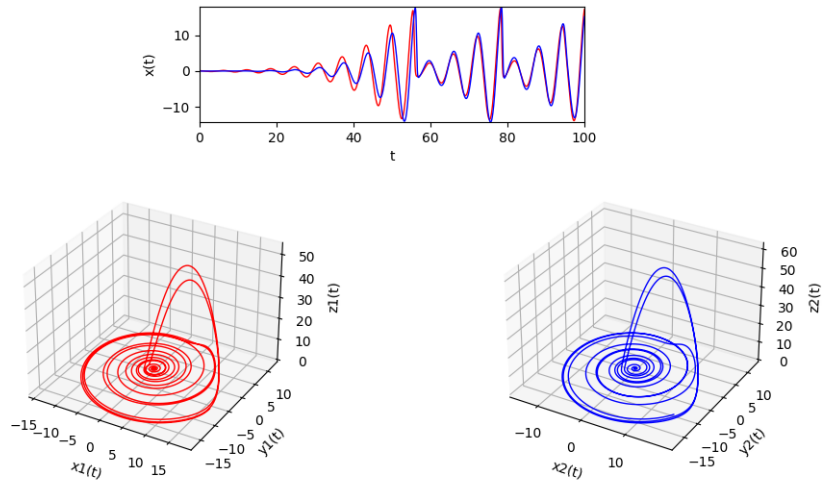


Рисунок 4.2: Два осциллятора, на примере которых будем рассматривать явление синхронизации.

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

$$\phi = \arctan\left(\frac{\dot{y}}{\dot{x}}\right); \quad (4.2)$$

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

$$v = \frac{d\phi}{dt} = \frac{\dot{y}\ddot{x} - \dot{x}\ddot{y}}{\dot{x}^2 + \dot{y}^2} \quad (4.3)$$

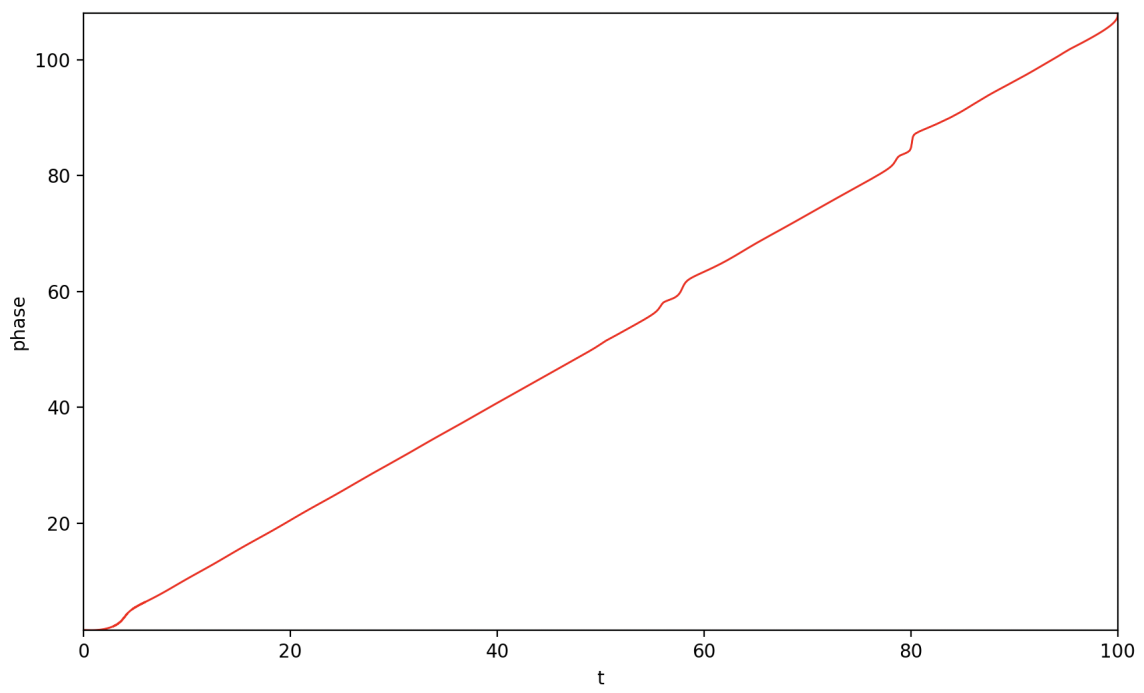


Рисунок 4.3: График зависимости фазы колебаний от времени. Монотонно возрастающая функция.

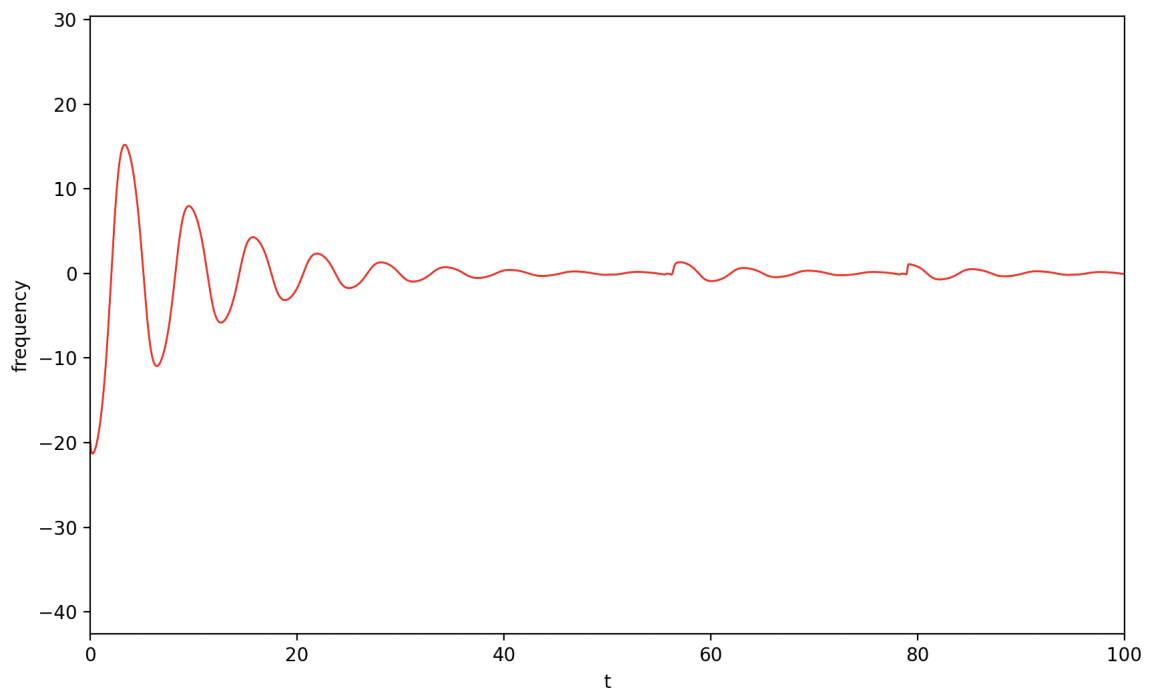


Рисунок 4.4: График зависимости мгновенной частоты от времени.

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 связи d :

$$\begin{cases} \frac{dx_{1,2}}{dt} = -w_{1,2}y_{1,2} - z_{1,2}, \\ \frac{dy_{1,2}}{dt} = w_{1,2}x_{1,2} + ay_{1,2} + d(y_{2,1} - y_{1,2}), \\ \frac{dz_{1,2}}{dt} = b + z_{1,2}(x_{1,2} - c). \end{cases} \quad (4.4)$$

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

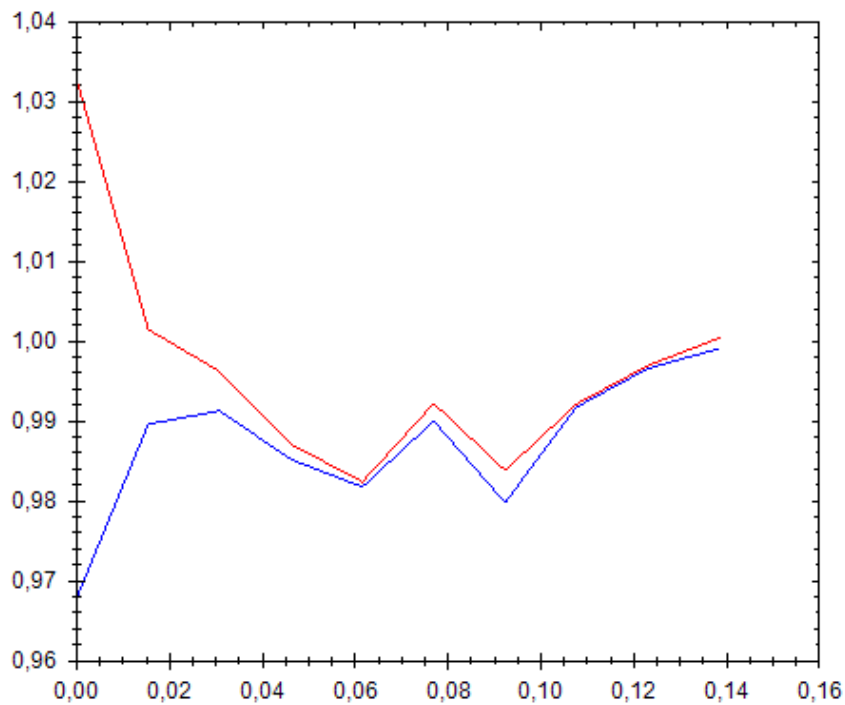


Рисунок 4.5: Синхронизация усредненных частот. Где d - коэффициент связи осцилляторов.

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.

$$\begin{cases} \frac{dx_i}{dt} &= -w_i y_i - z_i, \\ \frac{dy_i}{dt} &= w_i x_i + a y_i + \Sigma_j d(y_j - y_i), \\ \frac{dz_i}{dt} &= b + z_i(x_i - c). \end{cases} \quad (5.1)$$

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

$$d = \begin{cases} \bar{d} = \text{const}, & \text{if } (x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2 < r^2, \\ 0, & \text{otherwise.} \end{cases} \quad (5.2)$$

Пусть

1. $w_i \in [0, 93..1, 07]$.
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).

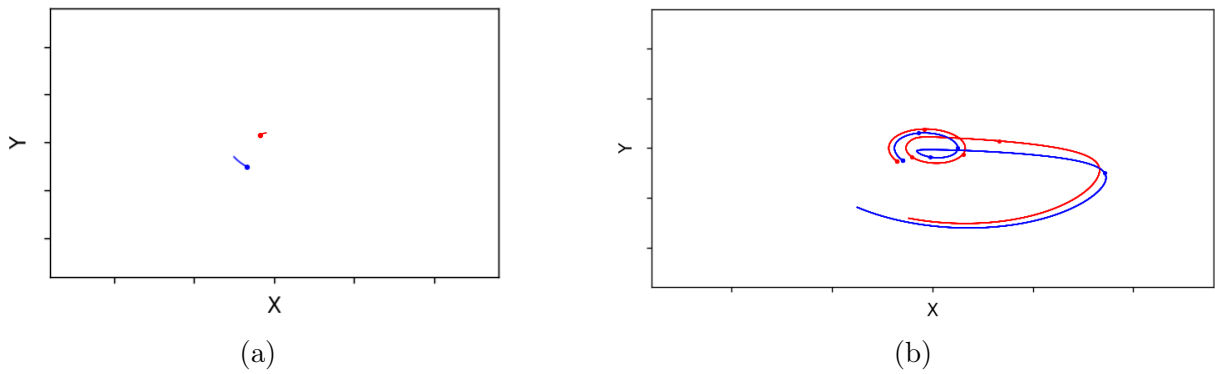


Рисунок 5.1: Расположение агентов на плоскости и след их движения

Увидеть момент синхронизации можно на следующем графике:

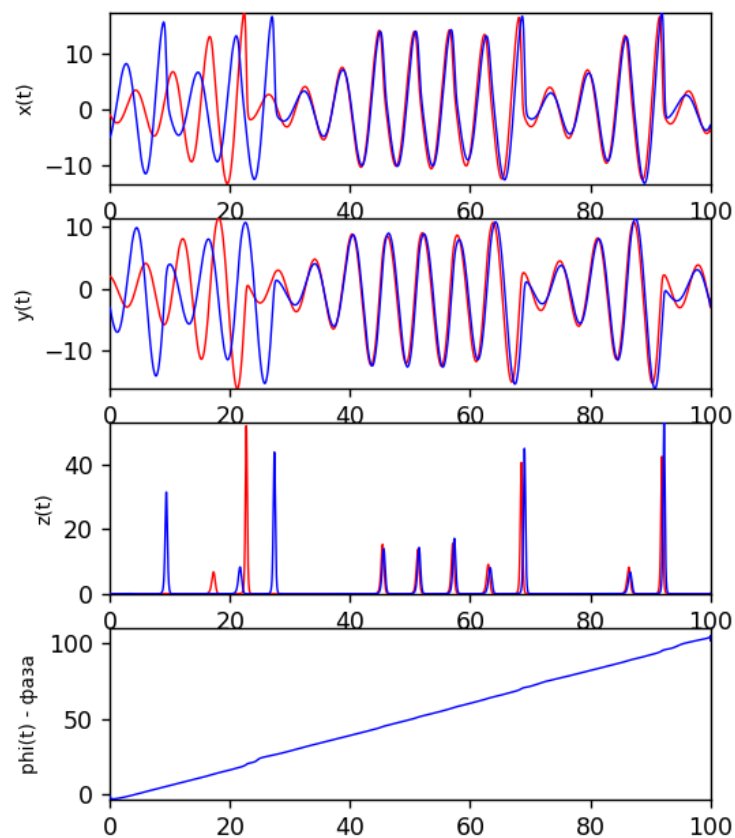


Рисунок 5.2: Изменения x , y , z в зависимости от времени. Наблюдения за процессом синхронизации.

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).

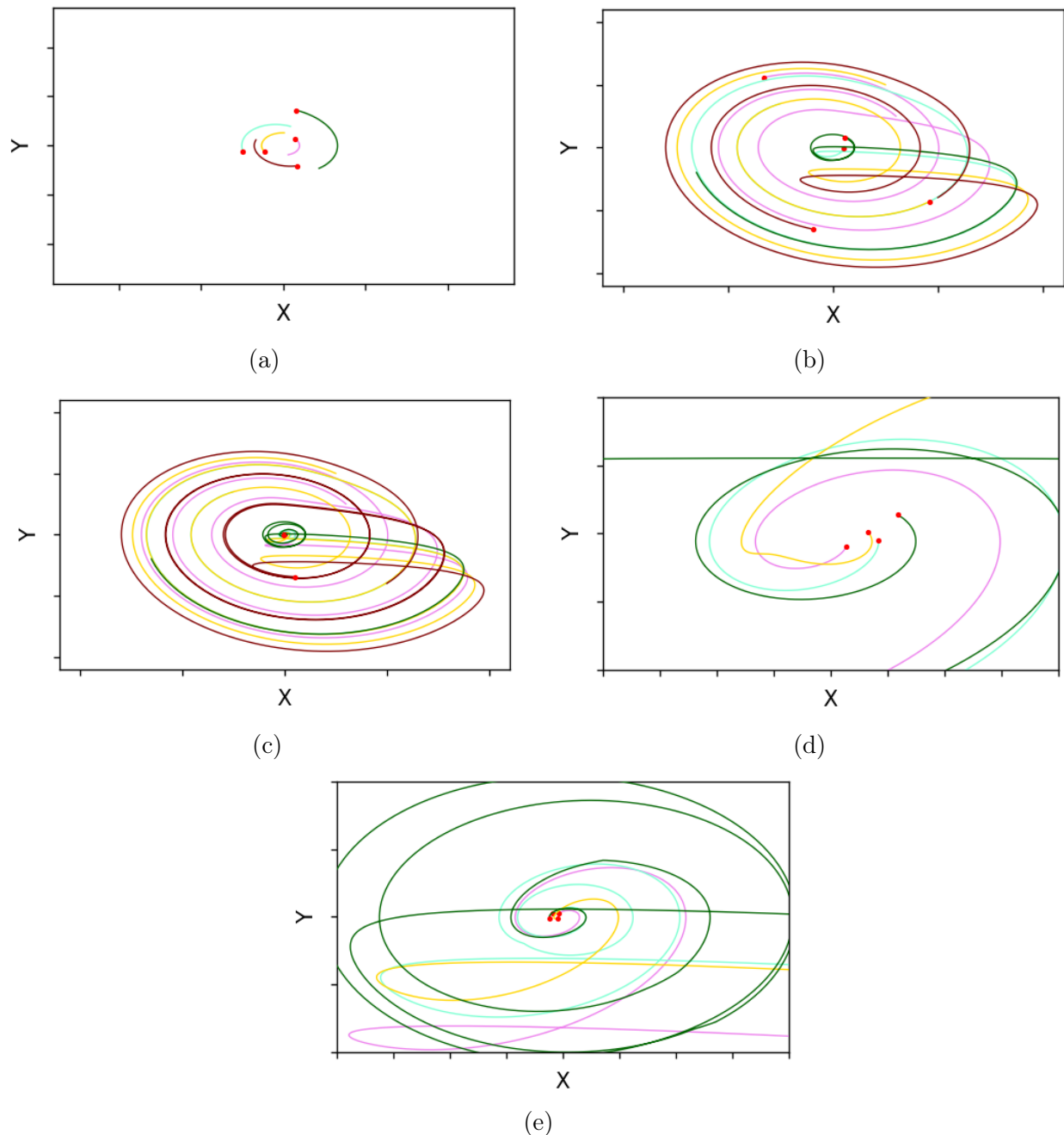


Рисунок 5.3: Расположение агентов на плоскости и след их движения (проекция на плоскость $\langle x, y \rangle$). Начальные точки выбраны случайным образом.

$$w_1 = 0.98, w_2 = 1, w_3 = 0.94, w_4 = 1.07, w_5 = 1.02$$

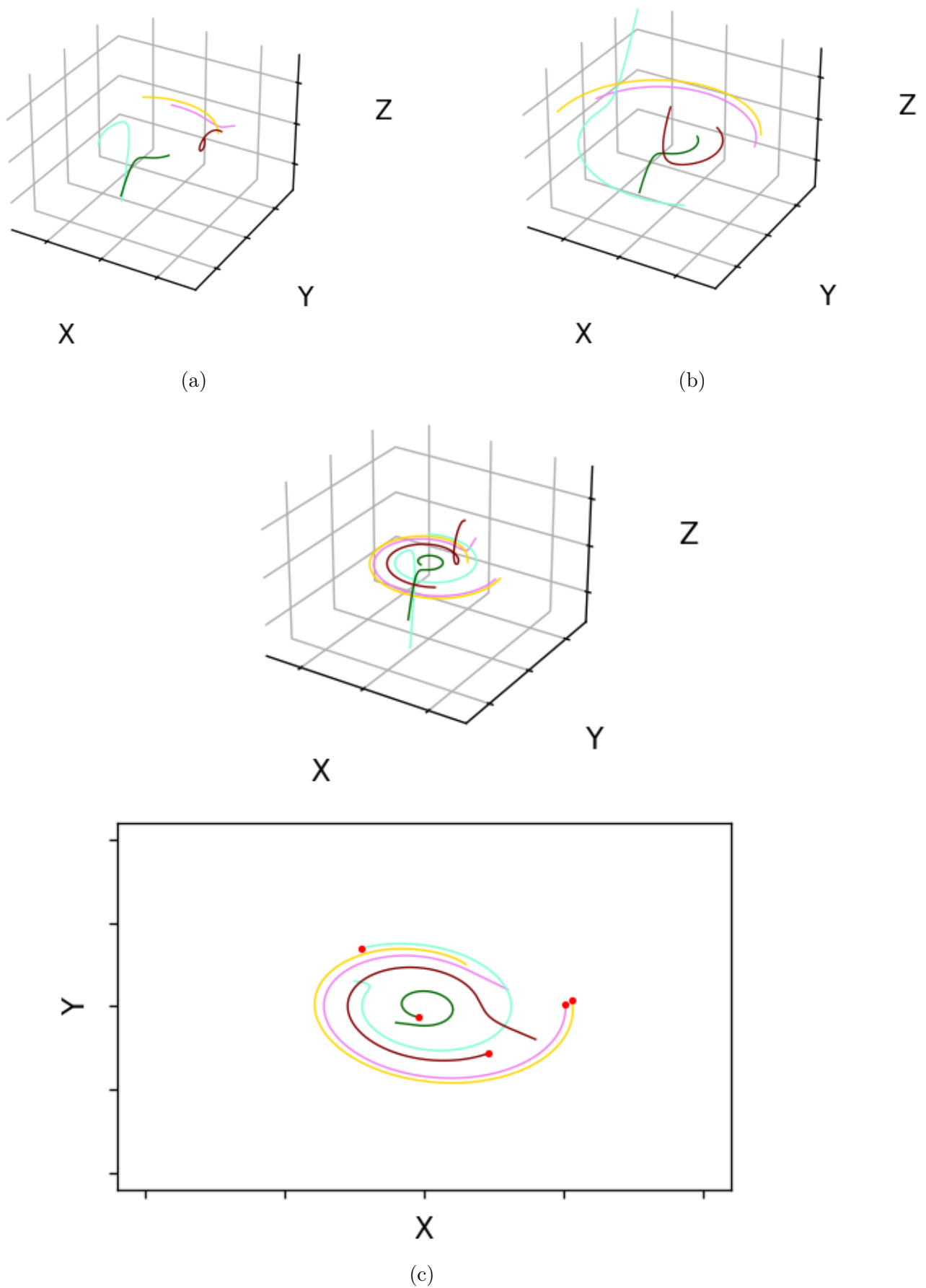


Рисунок 5.4: Расположение агентов на плоскости и след их движения (в пространстве).

Начальные точки выбраны случайным образом.

$w_1 = 0.98$, $w_2 = 1$, $w_3 = 0.94$, $w_4 = 1.07$, $w_5 = 1.02$,

а) начальное состояние системы, б) система спустя некоторое время, в) образование синхронизированных групп

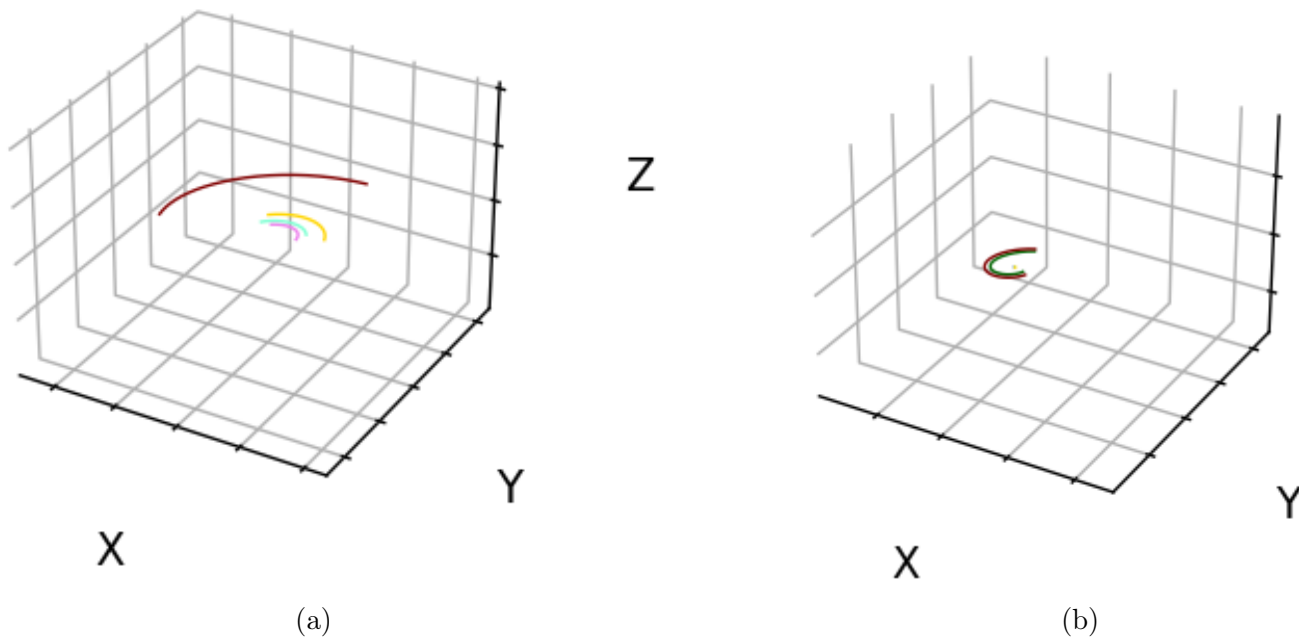


Рисунок 5.5: Синхронизация - локальная и глобальная
 а) синхронизация групп между собой, б) глобальная синхронизация

Увидеть момент синхронизации можно на следующих графиках:

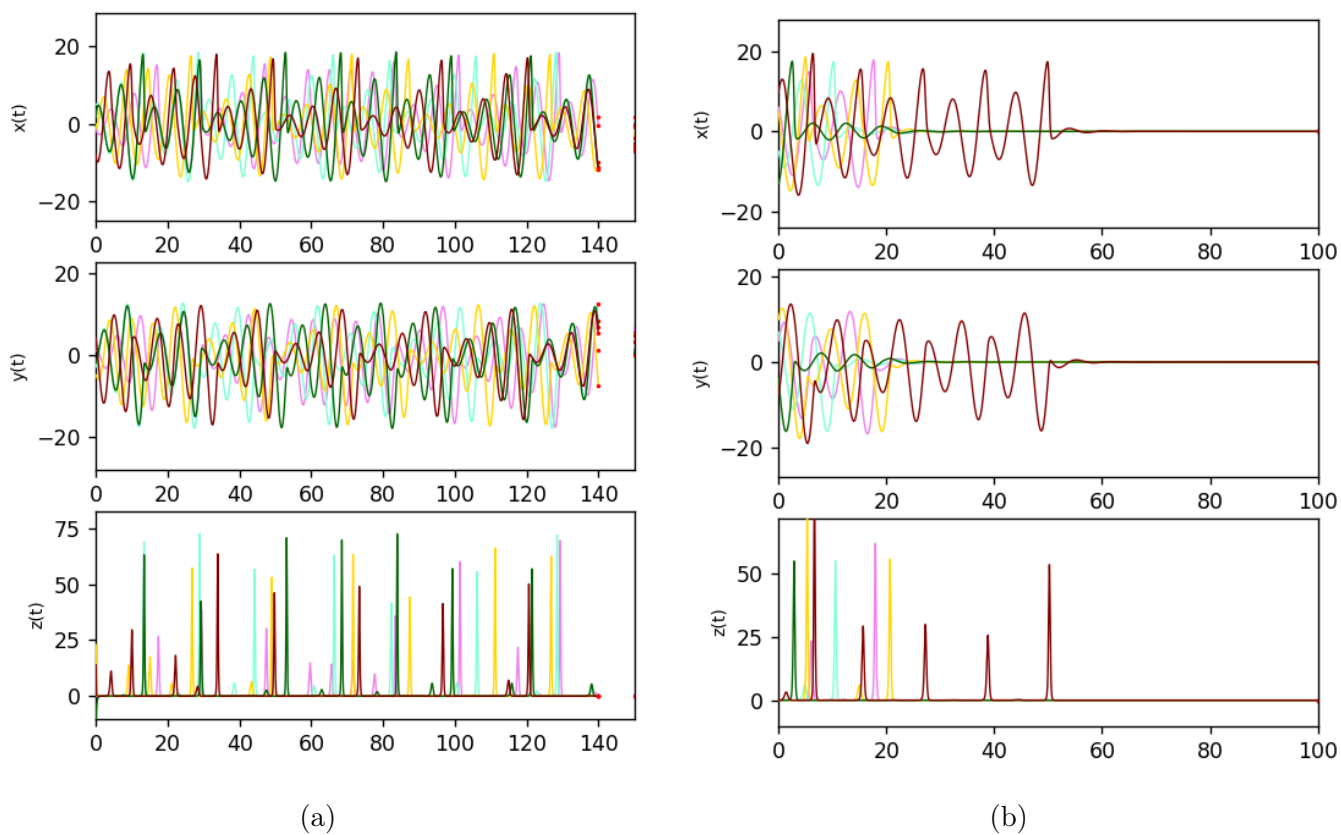


Рисунок 5.6: Синхронизация - локальная и глобальная

Рисунок 5.7: Изменения x , y , z в зависимости от времени. Наблюдения за процессом синхронизации. а) $d = 0.05$, б) $d = 0.2$ - вымирание колебаний

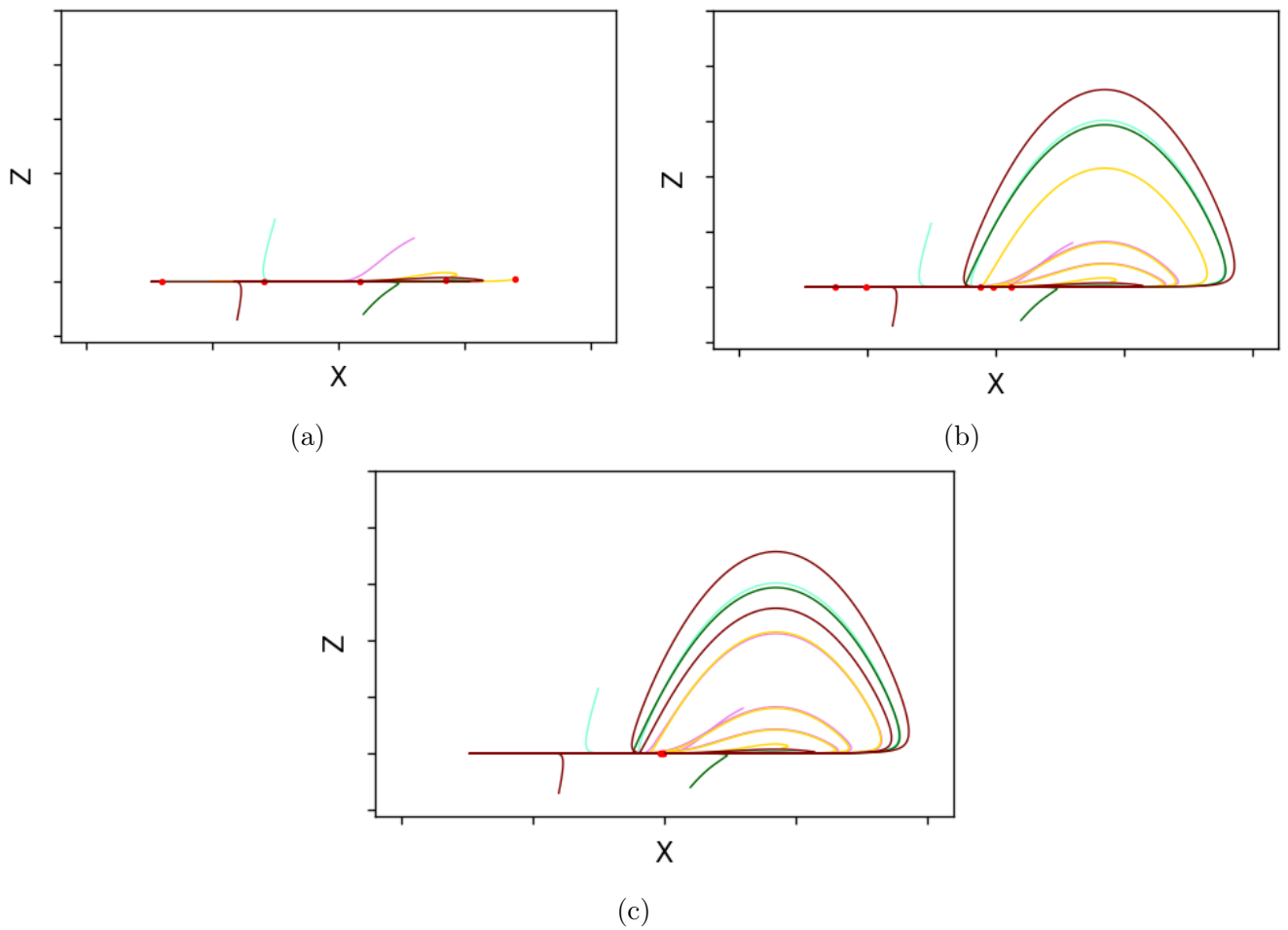


Рисунок 5.8: Расположение агентов на плоскости и след их движения (проекция на плоскость $\langle x, z \rangle$). Начальные точки выбраны случайным образом.
 $w_1 = 0.98$, $w_2 = 1$, $w_3 = 0.94$, $w_4 = 1.07$, $w_5 = 1.02$

В примере выше (Рис 5.7 - b) мы можем наблюдать такой эффект как осцилляторная смерть. Он возникает в том случае, если связь 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 "отталкивающую" связь по координате x :

$$\begin{cases} \frac{dx_i}{dt} = -w_i y_i - z_i + \sum_j d(x_j - x_i) + \sum_j d'/(x_i - x_j), \\ \frac{dy_i}{dt} = w_i x_i + a y_i, \\ \frac{dz_i}{dt} = b + z_i(x_i - c). \end{cases} \quad (6.1)$$

где

$$d, d' = \begin{cases} \bar{d} = const, if (x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2 < r^2, \\ 0, otherwise. \end{cases} \quad (6.2)$$

Эта связь работает следующим образом: при сближении i -го и j -го агентов $|d'/(x_i - x_j)|$ величина растёт. 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.

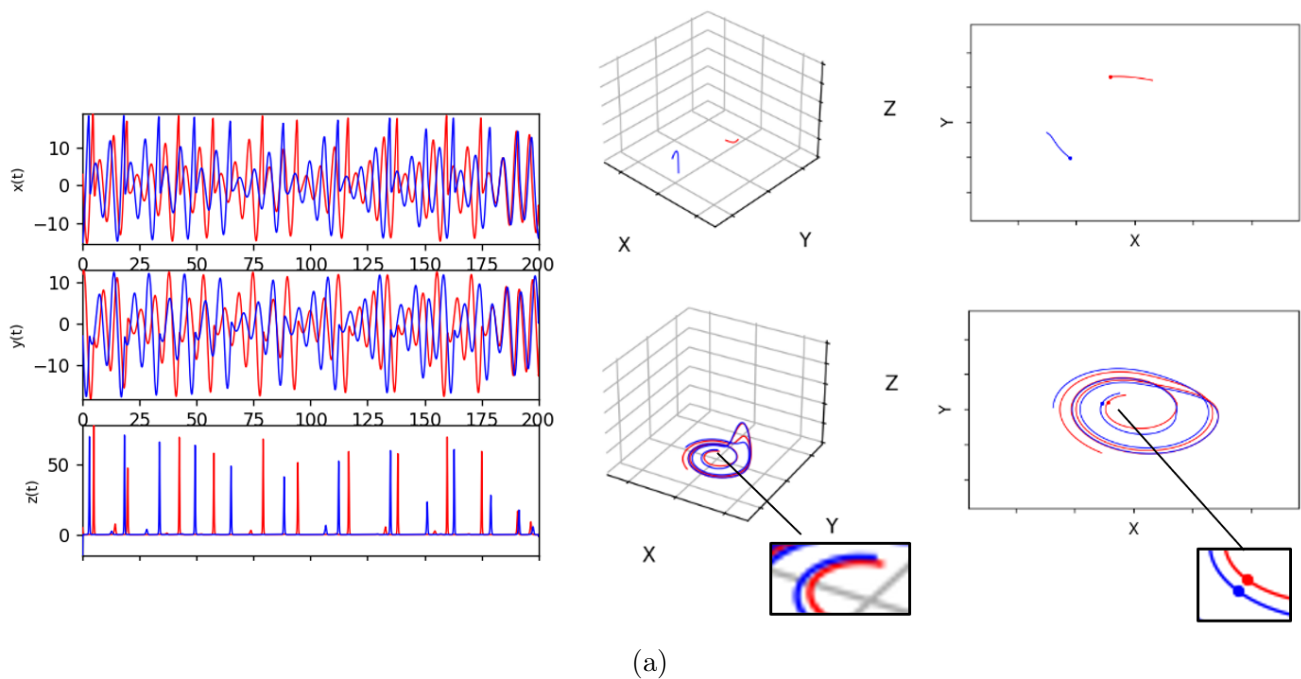


Рисунок 6.1: Расположение агентов на плоскости и след их движения, $N = 2$. Начальные точки выбраны случайным образом.

$w_1 = 0.98$, $w_2 = 1$, $w_3 = 0.94$, $w_4 = 1.07$, $w_5 = 1.02$

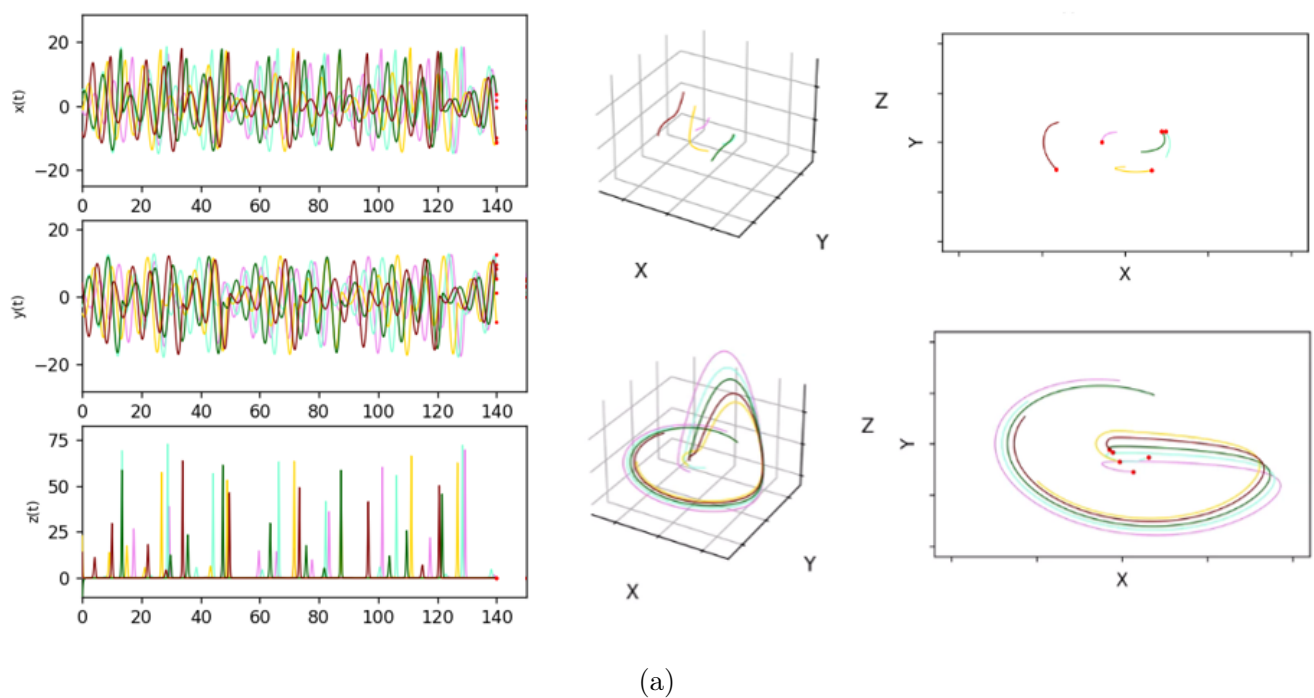


Рисунок 6.2: Расположение агентов на плоскости и след их движения, $N > 2$. Начальные точки выбраны случайным образом.

разница в значении параметра w около 0.01

Глава 7

Вывод синхронизированных агентов на траекторию, существенно отличающуюся от траектории синхронизации

В качестве целевой траектории возьмем хаотический Халворсен аттрактор:

$$\begin{cases} \frac{dX}{dt} = -aX - 4Y - 4Z - Y^2, \\ \frac{dY}{dt} = -aY - 4Z - 4X - Z^2, \\ \frac{dZ}{dt} = -aZ - 4X - 4Y - X^2. \end{cases} \quad (7.1)$$

где $a = 1.4$.

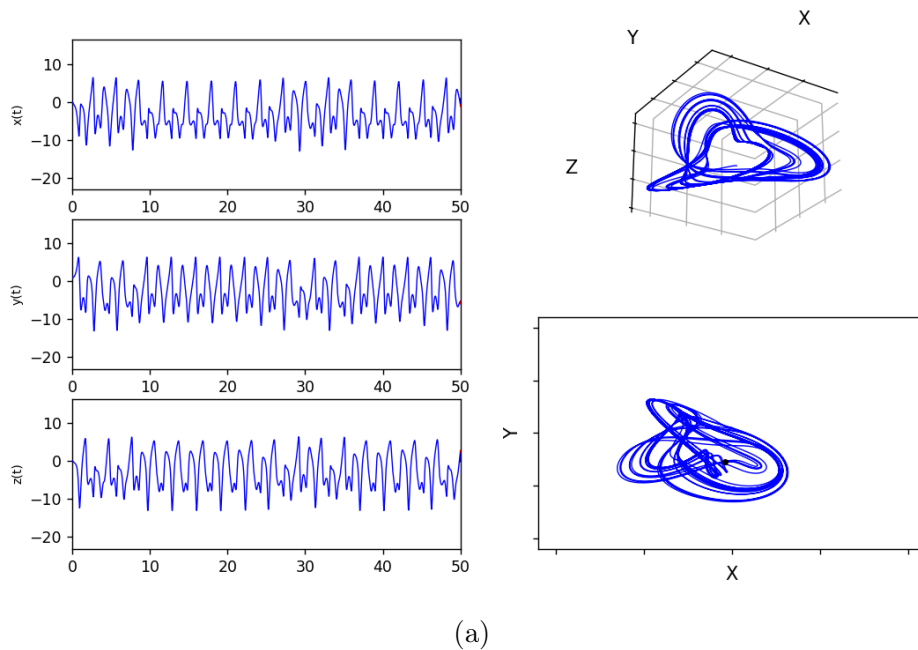


Рисунок 7.1: Целевая траектория - Халворсен аттрактор

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

$$\begin{cases} \frac{dx_i}{dt} = -w_i y_i - z_i + D(X - x_i), \\ \frac{dy_i}{dt} = w_i x_i + a y_i + \sum_j d(y_j - y_i) + D(Y - y_i), \\ \frac{dz_i}{dt} = b + z_i(x_i - c) + D(Z - z_i). \end{cases} \quad (7.2)$$

$$D = \begin{cases} \bar{d} = \text{const}, & \text{if } (x_i - X)^2 + (y_i - Y)^2 + (z_i - Z)^2 < r^2, \\ 0, & \text{otherwise.} \end{cases} \quad (7.3)$$

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

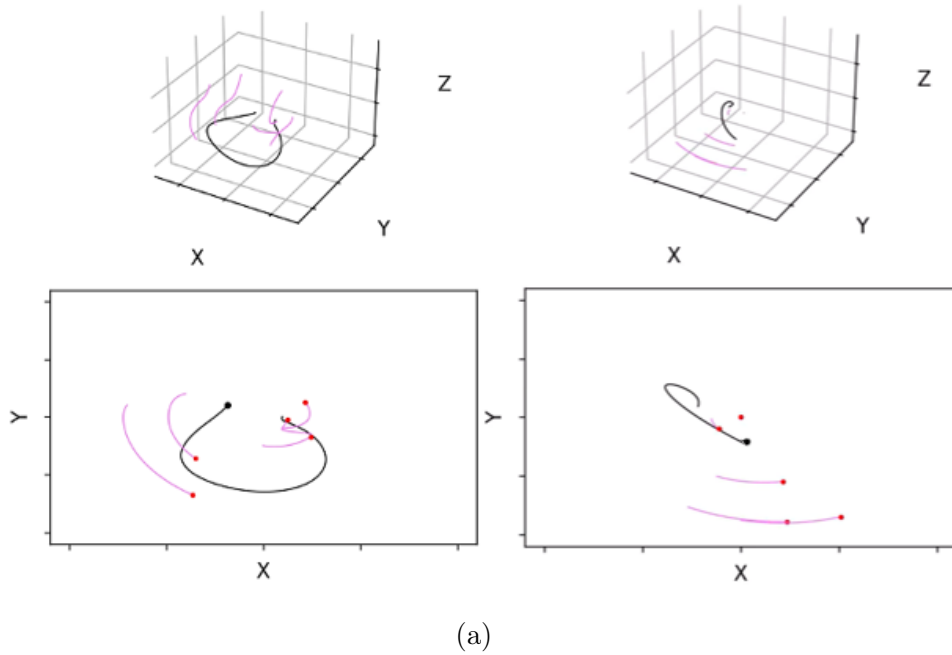


Рисунок 7.2: Расположение агентов и след их движения в пространстве. $N > 2$. Аттракторы Ресслера – розовый цвет, целевая траектория – чёрный.

Теперь в качестве целевой траектории возьмем хаотический аттрактор Лоренца:

$$\begin{cases} \frac{dX}{dt} = \sigma(Y - X), \\ \frac{dY}{dt} = X(\rho - Z) - Y, \\ \frac{dZ}{dt} = XY - \beta Z. \end{cases} \quad (7.4)$$

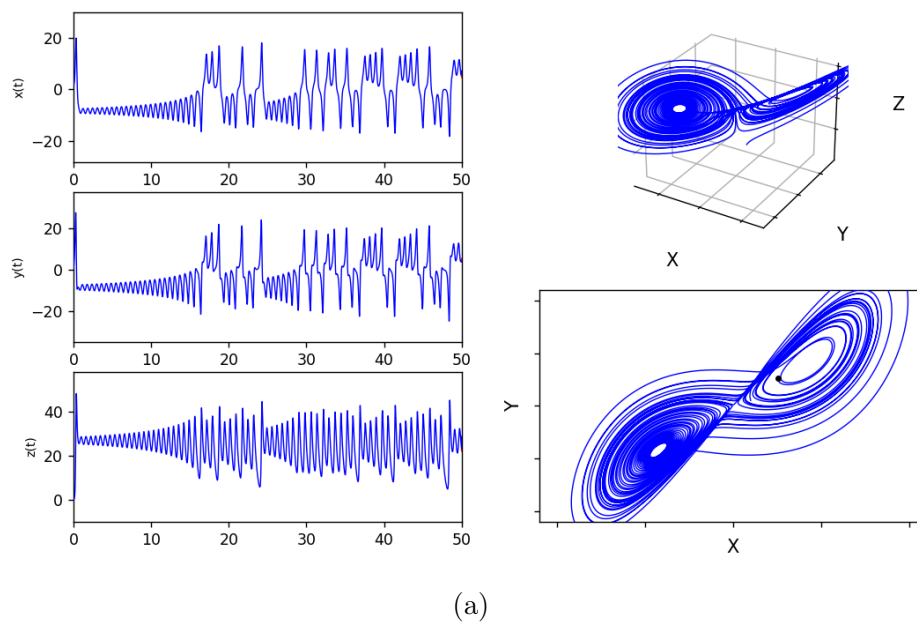


Рисунок 7.3: Целевая траектория - аттрактор Лоренца

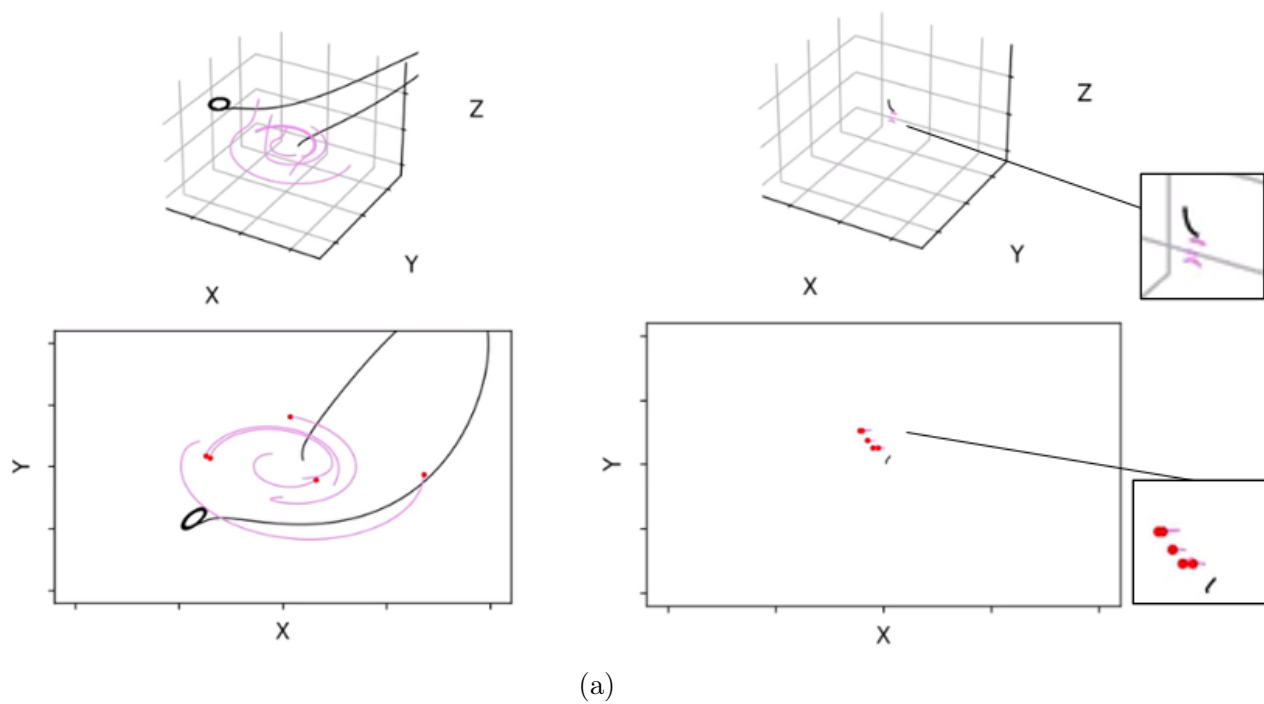


Рисунок 7.4: $N > 2$. Аттракторы Ресслера - розовый цвет, целевая траектория – чёрный.

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

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 from matplotlib.animation import FuncAnimation
5 from mpl_toolkits.mplot3d import Axes3D
6 from scipy.signal import hilbert
7
8 import math
9
10 def num_rossler(x1_n, y1_n, z1_n, h, a, b, c, w1, d_it):
11     def d(x1_n, x2_n, y1_n, y2_n):
12         d_val = 0
13         if ((x1_n - x2_n)**2 + (y1_n - y2_n)**2 < 3**2):
14             d_val = 0.2
15
16         return d_val
17
18     # --- connection : d_it = {0.2 or 0}
19
20     dx_1 = ((-w1) * y1_n - z1_n)
21     dy_1 = (w1 * x1_n + a * y1_n) # + d(x1_n, x2_n, y1_n, y2_n) * (y2_n - y1_n)
22     dz_1 = (b + z1_n * (x1_n - c))
23
24     x1_n1 = x1_n + h * dx_1
25     y1_n1 = y1_n + h * dy_1
26     z1_n1 = z1_n + h * dz_1
27
28     return x1_n1, y1_n1, z1_n1
```

```

29
30 def num_rossler_freq(x1_n, y1_n, z1_n, h, a, b, c, w1, d_it):
31     def d(x1_n, x2_n, y1_n, y2_n):
32         d_val = 0
33         if ((x1_n - x2_n)**2 + (y1_n - y2_n)**2 < 3**2):
34             d_val = 0.2
35             # d_val = d_it
36
37         return d_val
38
39     # --- connection : d_it = {0.2 or 0}
40
41     dx_1 = ((-w1) * y1_n - z1_n)
42     dy_1 = (w1 * x1_n + a * y1_n) # + d(x1_n, x2_n, y1_n, y2_n) * (y2_n - y1_n)
43     dz_1 = (b + z1_n * (x1_n - c))
44
45     dx_1_v = ((-w1) * y1_n - z1_n)
46     dy_1_v = (w1 * x1_n + a * y1_n)
47     dz_1_v = (b + z1_n * (x1_n - c))
48
49     x1_n1 = x1_n + h * dx_1
50     y1_n1 = y1_n + h * dy_1
51     z1_n1 = z1_n + h * dz_1
52
53     phi_1 = np.arctan(dy_1/dx_1)
54     phi_1 = np.arctan(y1_n/x1_n)
55
56     dphi_1 = ((w1 + a) * dx_1 + (-w1 - 1) * dy_1) / (dx_1**2)
57
58     dphi_f_1 = 1 / (1 + (dy_1/dx_1)**2)
59     dphi_g_1 = dphi_1
60     res_dphi1 = dphi_f_1 * dphi_g_1
61
62     d_dx = (-1)*w1 - 1
63     d_dy = w1 + a
64
65     first = dy_1_v * d_dx - dx_1_v * d_dy
66     sec = dx_1_v**2 + dy_1_v**2
67
68     res_dphi1 = (first/sec)
69
70     phase_diff_1 = phi_1
71

```

```

72     return res_dphi1, phase_diff_1
73
74
75 def rossler_eq(x, y, z, a, b, c, w):
76     #
77     dx = (-w) * y - z
78     dy = w * x + a * y
79     dz = b + z * (x - c)
80
81     phi_1 = np.arctan(dy/dx)
82
83     d_dx = (-1)*w - 1
84     d_dy = w + a
85
86     first = dy * d_dx - dx * d_dy
87     sec = dx**2 + dy**2
88
89     res_dphi1 = first / sec
90
91     return dx, dy, dz, res_dphi1
92
93
94 def solve(x, y, z, phi, v, t, a, b, c, w):
95     for i in range(len(t) - 1):
96         dx, dy, dz, dphi = rossler_eq(x[i], y[i], z[i], a, b, c, w)
97
98         dt = t[i + 1] - t[i]
99
100        phi[i] = np.arctan(dy / dx)
101        v[i] = dphi
102
103        x[i + 1] = x[i] + (dx * dt)
104        y[i + 1] = y[i] + (dy * dt)
105        z[i + 1] = z[i] + (dz * dt)
106
107 def visualize(x1, y1, z1, t, freq_1, time_frame, steps_t, steps_fr, N):
108     fig = plt.figure(figsize=(10, 6))
109
110     h1 = hilbert(x1[0])
111     phi1 = np.unwrap(np.angle(h1))
112
113     col = (np.random.random(), np.random.random(), np.random.random())
114
115     axis_1 = fig.add_subplot(2, 2, 2, projection='3d')

```

```

116     line1, = axis_1.plot([], [], [], color=col, linewidth=0.8) # osc
117
118     axis_1_track = fig.add_subplot(2, 2, 4)
119     line1_track, = axis_1_track.plot([], [], color=col, linewidth=0.8) # osc
120
121     point_track1, = axis_1_track.plot([], [], marker='o', color=col, markersize
        =2)
122
123     point, = axis_1.plot([], [], [], marker='o', color=col, markersize=1)
124
125     axis_1.xaxis.set_pane_color((1.0, 0.0, 1.0, 0.0))
126     axis_1.yaxis.set_pane_color((1.0, 0.0, 1.0, 0.0))
127     axis_1.zaxis.set_pane_color((1.0, 0.0, 1.0, 0.0))
128
129     axis_1.set_xlim(-28, 28)
130     axis_1.set_ylim(-28, 28)
131     axis_1.set_zlim(0, 50)
132
133     axis_1_track.set_xlim(-28, 28)
134     axis_1_track.set_ylim(-28, 28)
135
136     axis_1.set_xlabel('X', fontsize=12)
137     axis_1.set_ylabel('Y', fontsize=12)
138     axis_1.set_zlabel('Z', fontsize=12)
139
140     axis_1_track.set_xlabel('X', fontsize=12)
141     axis_1_track.set_ylabel('Y', fontsize=12)
142
143     axis_1.set_yticklabels([])
144     axis_1.set_xticklabels([])
145     axis_1.set_zticklabels([])
146
147     axis_1_track.set_yticklabels([])
148     axis_1_track.set_xticklabels([])
149
150     axis_2_x = fig.add_subplot(4, 2, 1)
151     line3_1_x, = axis_2_x.plot([], [], 'b-', linewidth=0.8) # osc 1
152
153     point3_1_x, = axis_2_x.plot([], [], marker='o', color=col, markersize=1)
154
155     axis_2_y = fig.add_subplot(4, 2, 3)
156     line3_1_y, = axis_2_y.plot([], [], 'b-', linewidth=0.8) # osc 1
157
158     point3_1_y, = axis_2_y.plot([], [], marker='o', color=col, markersize=1)

```

```

159
160 axis_2_z = fig.add_subplot(4, 2, 5)
161 line3_1_z, = axis_2_z.plot([], [], 'b-', linewidth=0.8) # osc 1
162
163 point3_1_z, = axis_2_z.plot([], [], marker='o', color=col, markersize=1)
164
165 axis_3 = fig.add_subplot(4, 2, 7)
166 line_phi, = axis_3.plot([], [], 'g-', linewidth=0.8)
167
168 axis_3.set_xlim(0, time_frame)
169 axis_3.set_ylim(min(phi1), max(phi1))
170 axis_3.set_ylabel('phi(t) - ', fontsize=8)
171 point_phi, = axis_3.plot([], [], marker='o', color='b', markersize=1)
172
173 # --- setting the data
174
175 axis_2_x.set_xlim(t[0], t[len(t) - 1])
176 axis_2_x.set_xlim(0, time_frame)
177 axis_2_x.set_ylim(min(x1[0]) - 10, max(x1[0] + 10))
178 axis_2_x.set_ylabel('x(t)', fontsize=8)
179
180 axis_2_y.set_xlim(t[0], t[len(t) - 1])
181 axis_2_y.set_xlim(0, time_frame)
182 axis_2_y.set_ylim(min(y1[0]) - 10, max(y1[0]) + 10)
183 axis_2_y.set_ylabel('y(t)', fontsize=8)
184
185 axis_2_z.set_xlim(t[0], t[len(t) - 1])
186 axis_2_z.set_xlim(0, time_frame)
187 axis_2_z.set_ylim(min(z1[0]) - 10, max(z1[0]) + 10)
188 axis_2_z.set_ylabel('z(t)', fontsize=8)
189
190 # when_to_track = round((steps_t/10)*9)
191 # when_to_track = 0
192
193 # x1_val_track = x1[when_to_track:steps_t]
194 # y1_val_track = y1[when_to_track:steps_t]
195
196 colors = ['violet',
197           'aquamarine',
198           'gold',
199           'darkgreen',
200           'maroon']
201
202 def update_all(i):

```

```

203     if (i < steps_t):
204         x1_val = x1[n][0:i]
205         y1_val = y1[n][0:i]
206         z1_val = z1[n][0:i]
207
208         tq = t[0:i]
209
210         v1_val = freq_1[n][0:i]
211
212         line1.set_data(x1_val, y1_val)
213         line1.set_3d_properties(z1_val)
214         line1.set_3d_properties(z1_val)
215
216         point.set_3d_properties(z1[n][i])
217         line1.set_color(colors[n])
218
219         line1_track.set_color(colors[n])
220
221         #----- x pro
222
223         line3_1_x.set_data(tq, x1_val)
224         point3_1_x.set_data(t[i], x1[n][i])
225         line3_1_x.set_color(colors[n])
226
227         #----- y pro
228
229         line3_1_y.set_data(tq, y1_val)
230         point3_1_y.set_data(t[i], y1[n][i])
231         line3_1_y.set_color(colors[n])
232
233         #----- z pro
234
235         line3_1_z.set_data(tq, z1_val)
236         point3_1_z.set_data(t[i], z1[n][i])
237         line3_1_z.set_color(colors[n])
238
239     # --- from the when_to_track point
240     # if (i > when_to_track):
241         # line1_track.set_data(x1_val_track, y1_val_track)
242         # line2_track.set_data(x2_val_track, y2_val_track)
243         # line1_track.set_data(x1_val, y1_val)
244         # line2_track.set_data(x2_val, y2_val)
245
246         # point_track1.set_data(x1[i], y1[i])

```

```

247         # point_track2.set_data(x2[i], y2[i])
248
249     # --- from the start
250     line1_track.set_data(x1_val, y1_val)
251
252     point_track1.set_data(x1[n][i], y1[n][i])
253     return line1, line1_track, line3_1_x, line3_1_y, line3_1_z, point,
254         point_track1, point3_1_x, point3_1_y, point3_1_z
255
256 def update_phi(j):
257     j = (j - 1) * 100 - 1
258
259     if (j < steps_t & j < steps_fr):
260         tq = t[0:j]
261
262         phi_val = phi1[0:j]
263
264         line_phi.set_data(tq, phi_val)
265         point_phi.set_data(t[j], phi1[j])
266         line_phi.set_color("blue")
267
268     return line_phi, point_phi
269
270 anim_phi = FuncAnimation(fig, update_phi, frames = np.size(x1[0]), interval
271     = 0, blit = True)
272
273 anim = FuncAnimation(fig, update_all, frames = np.size(x1[0]), interval =
274     0, blit = True)
275
276 plt.show()
277 plt.savefig('rosslerAttractor.png')
278
279
280 def main():
281     N_of_agents = 5
282
283     a = 0.22
284     b = 0.1
285     c = 8.5
286
287     t_ini = 0
288     t_fin = 100
289     h = 0.0001
290     numsteps = int((t_fin - t_ini) / h)
291     t = np.linspace(t_ini, t_fin, numsteps)

```



```

288
289     d_ini = 0
290     d_fin = 0.16
291     d_h = 0.00000016
292     d_numsteps = int((d_fin - d_ini) / d_h)
293     d = np.linspace(d_ini, d_fin, d_numsteps)
294
295     x1 = np.random.rand(N_of_agents, numsteps)
296     y1 = np.random.rand(N_of_agents, numsteps)
297     z1 = np.random.rand(N_of_agents, numsteps)
298
299     freq_1 = np.random.rand(N_of_agents, numsteps)
300     phase_diff_1 = np.random.rand(N_of_agents, numsteps)
301
302     for i in range(N_of_agents):
303         w1=np.random.uniform(0.93,1.07) # random in the range of (0,93 .. 1,07)
304
305         x1[i] = np.zeros(numsteps)
306         y1[i] = np.zeros(numsteps)
307         z1[i] = np.zeros(numsteps)
308
309         freq_1[i] = np.zeros(numsteps)
310         phase_diff_1[i] = np.zeros(numsteps)
311
312         # random start point
313         x1[i][0] = np.random.uniform(-5, 5)
314         y1[i][0] = np.random.uniform(-5, 5)
315         z1[i][0] = np.random.uniform(-5, 5)
316
317         for k in range(numsteps-1):
318             [x1[i][k+1], y1[i][k+1], z1[i][k+1]] = num_rossler(x1[i][k], y1[i][
319                 k], z1[i][k], t[k+1] - t[k], a, b, c, w1, d[k])
320
321         for k in range(d_numsteps-1):
322             [freq_1[i][k+1], phase_diff_1[i][k+1]] = num_rossler_freq(x1[i][k],
323                 y1[i][k], z1[i][k], t[k+1]-t[k], a, b, c, w1, d[k])
324
325     visualize(x1, y1, z1, t, freq_1, t_fin, numsteps, d_numsteps, N_of_agents)
326
327 if __name__ == "__main__":
328     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 системы Ресслер. 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, China 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 Rössler 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