МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Д-р. техн. наук, доцент |  |  |  | С. И. Колесникова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №5 |
| Моделирование линейных/нелинейных объектов. Линейные системы. Передаточные функции. Модели детерминированного хаоса. Режимы устойчивости/неустойчивости. Автоколебательные модели. |
| по курсу: компьютерное моделирование |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ гр. № | 4134 |  |  |  | Самарин Д. В. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2025

1. **Цель работы**

Цель настоящей работы: знакомство с элементами синергетического управления применительно к моделям детерминированного хаоса, с принципами организация обратных связей в сложных объектах для достижения режима устойчивости функционирования нелинейного объекта.

1. **Задание на лабораторную работу**

Часть 1.

1. Ознакомиться со справочными сведениями.

2. Построить графики и фазовые портреты нелинейной модели для устойчивого и неустойчивого режимов.

3. Разработать программу, реализующую алгоритм управления хаотической моделью с целью стабилизации объекта в окрестности устойчивого состояния.

4. Получить сравнительные графики управляемой и неуправляемой моделей.

5. Составить и представить преподавателю отчет о работе.

Часть 2.

1. Ознакомиться со справочными сведениями относительно применения дискретных/непрерывных блоков Simulink.

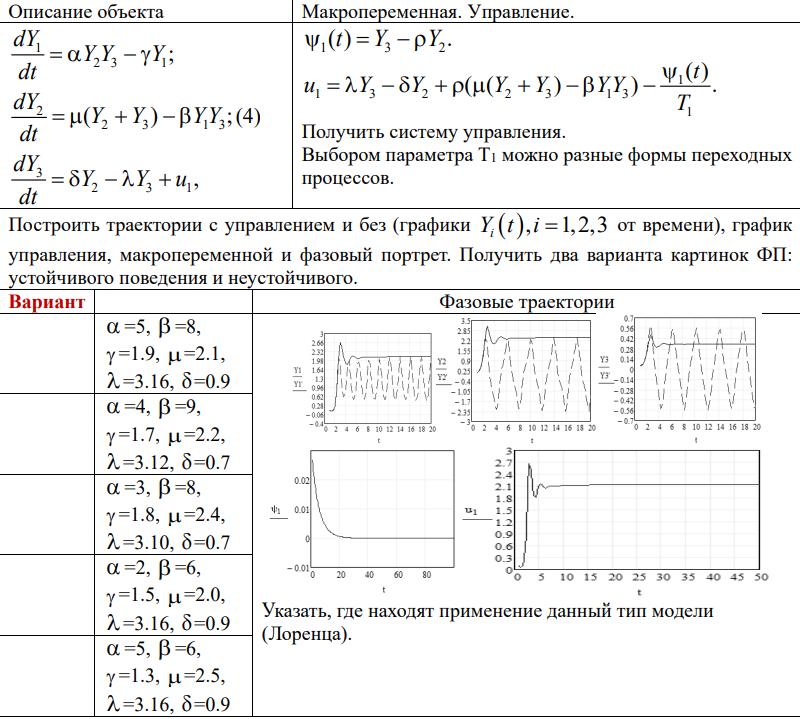
2. Построить модель системы автоматического регулирования в Simulink.

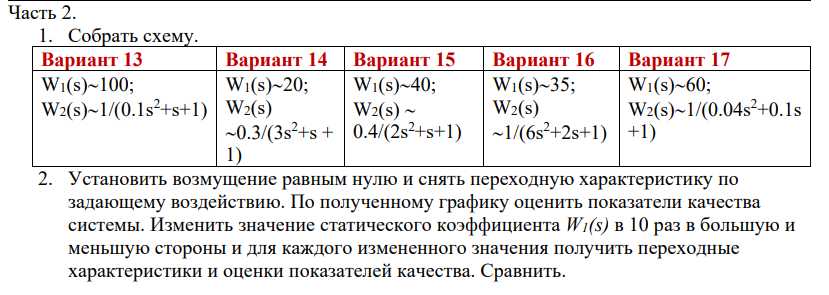
3. В отчет включить схему и скриншоты окон настроек каждого блока.

4. Описать принцип работы блока Линейные системы.

5. Представить необходимые графики

**Вариант: 14**





1. **Ход работы:**

**Часть 1:**

Проведём детальное исследование описания системы:

**Описание объекта**

Уравнения описания объекта описывают трехмерную динамическую систему с нелинейными взаимодействиями между , которая подвержена внешнему управлению через u1.

зависит от нелинейного взаимодействия , масштабированных с помощью , а также от собственных потерь с учетом коэффициента

зависит от линейного взаимодействия , масштабированных , и от нелинейного взаимодействия , масштабированных .

линейно зависит от с коэффициентом , собственного затухания, которое определяется и, собственно, внешнего управления .

**Макропеременная**

Макропеременная описывает отклонение системы от её равновесного состояния. Коэффициент определяет значимость отдельных составляющих относительно состояния:

Она служит индикатором текущего состояния системы, оказывая влияние на управление и корректируя процесс стабилизации. Функция макропеременной детализирована в последующих разделах.

**Функция управления**

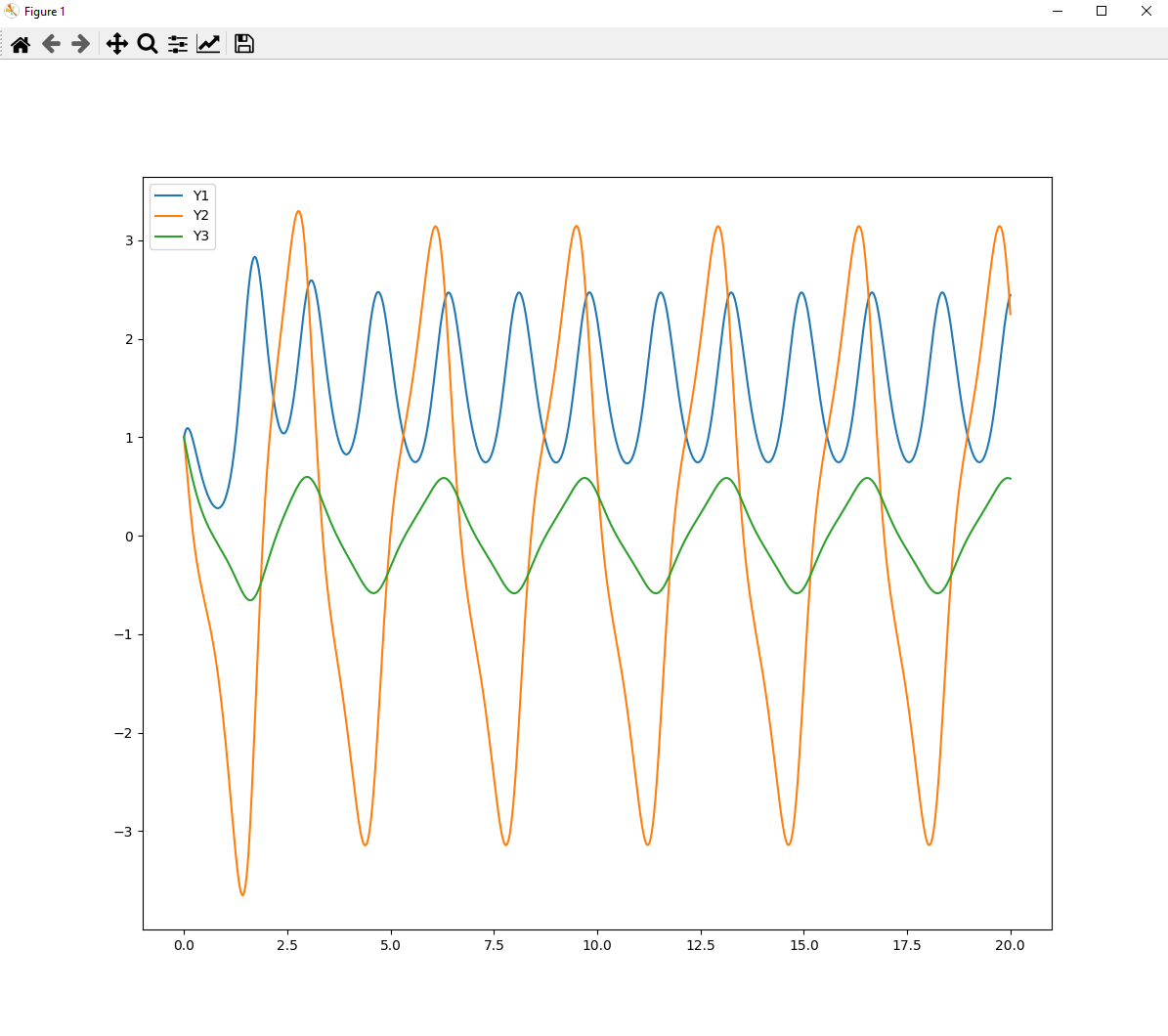
Для стабилизации системы применяется корректирующее управление. Основу этой функции составляет параметр, регулирующий скорость реакции системы:

Управление активно стабилизирует состояние. Значение определяет баланс между быстродействием и плавностью переходного процесса.

**Анализ состояния без управления**

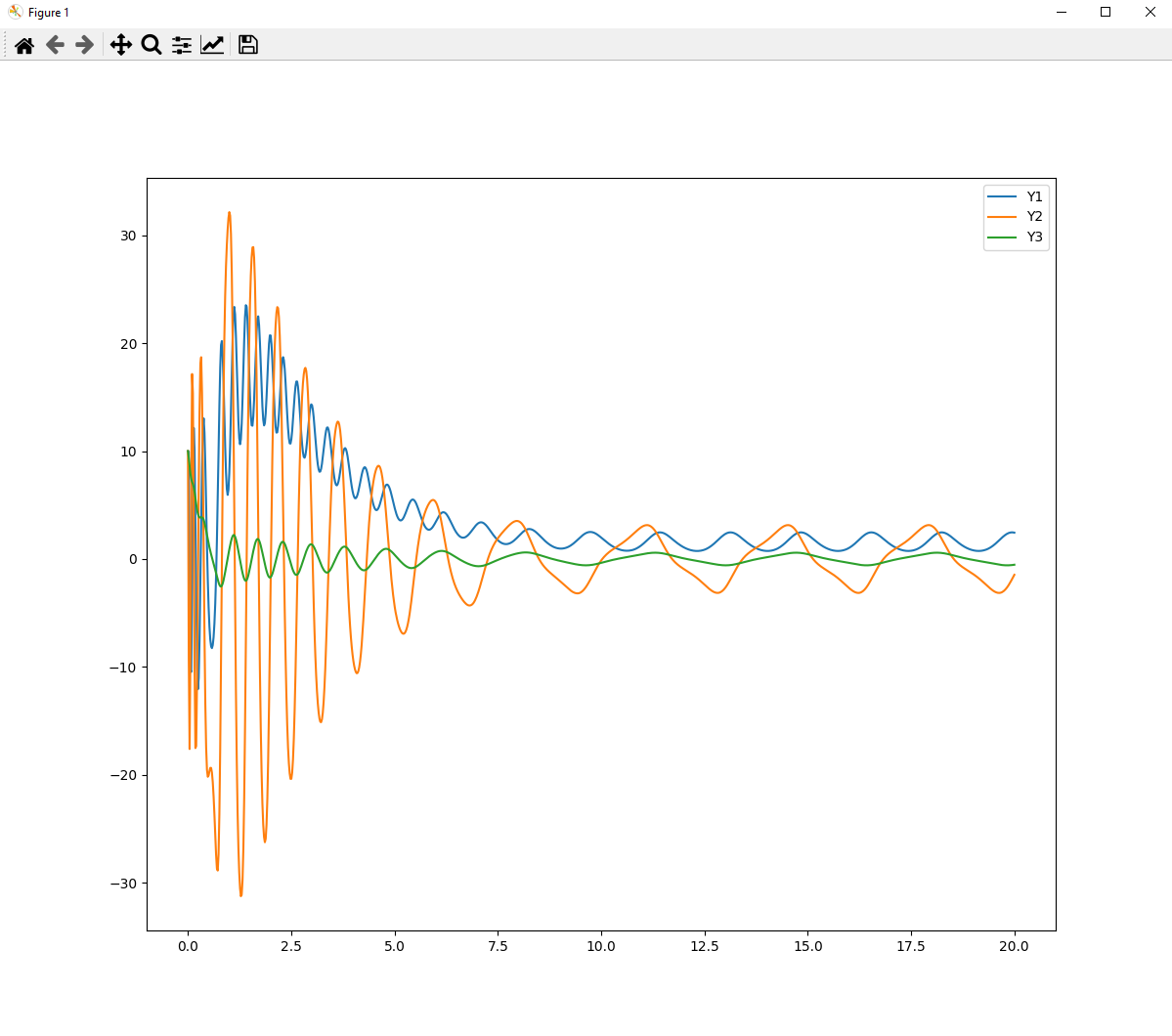
Предположим, что начальные значения всех переменных системы равны 1. В отсутствие управления динамика будет характеризоваться исключительно внутренними взаимодействиями и затуханием

:



При заданных начальных значениях и параметрах системы каждая переменная демонстрирует поведение, характерное для режима граничного цикла. Это указывает на наличие устойчивых периодических колебаний с постоянной амплитудой. Такая динамика обусловлена сбалансированным влиянием обратных связей между переменными, которые удерживают систему в стационарном колебательном режиме.

:

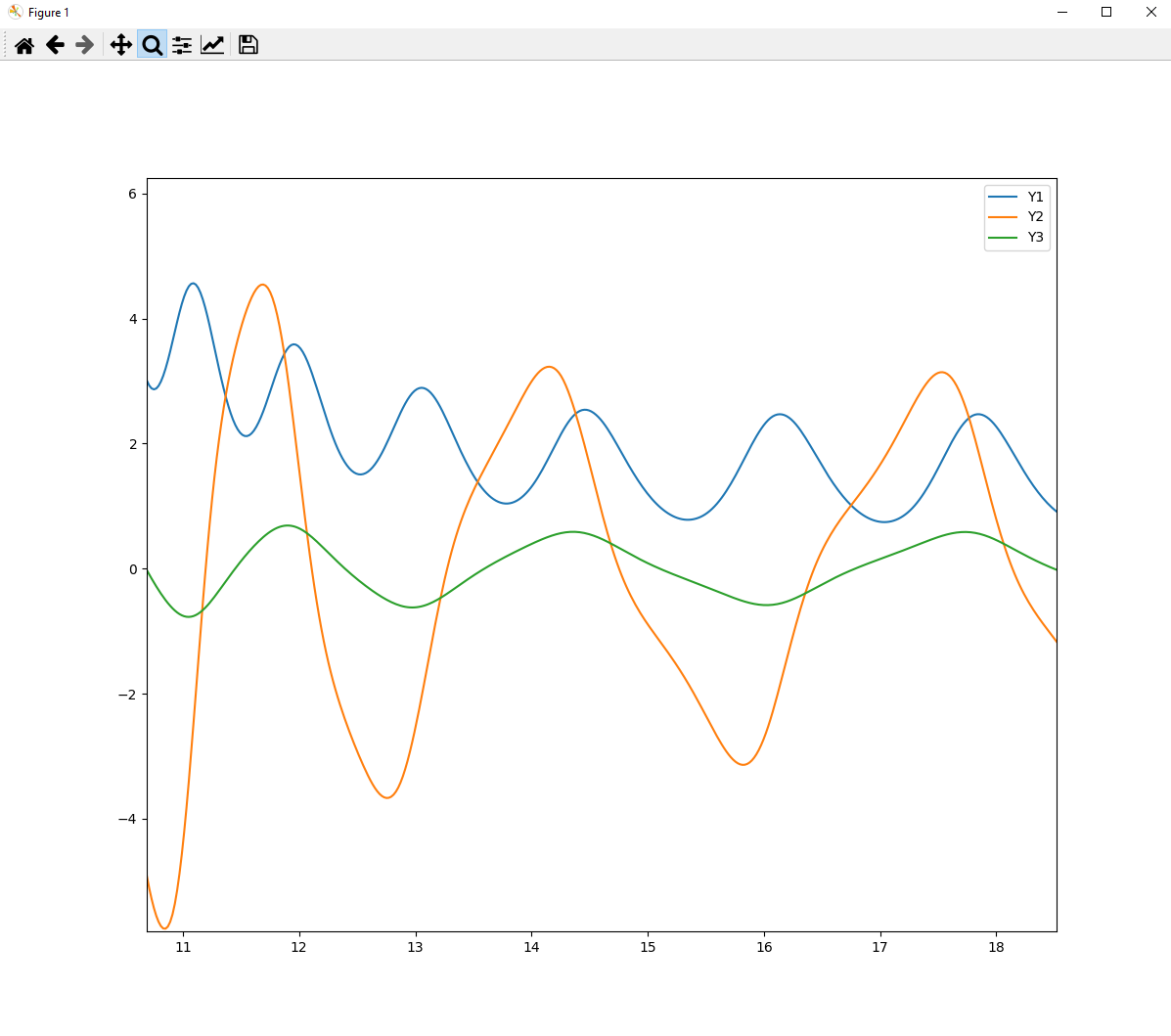


При заданных начальных условиях система демонстрирует признаки частичной устойчивости: амплитуда колебаний ее переменных постепенно уменьшается. Со временем значения этих переменных приходят в состояние равновесия в пределах той же области, что и при .

Дабы убедиться в верности наших рассуждений возьмем .

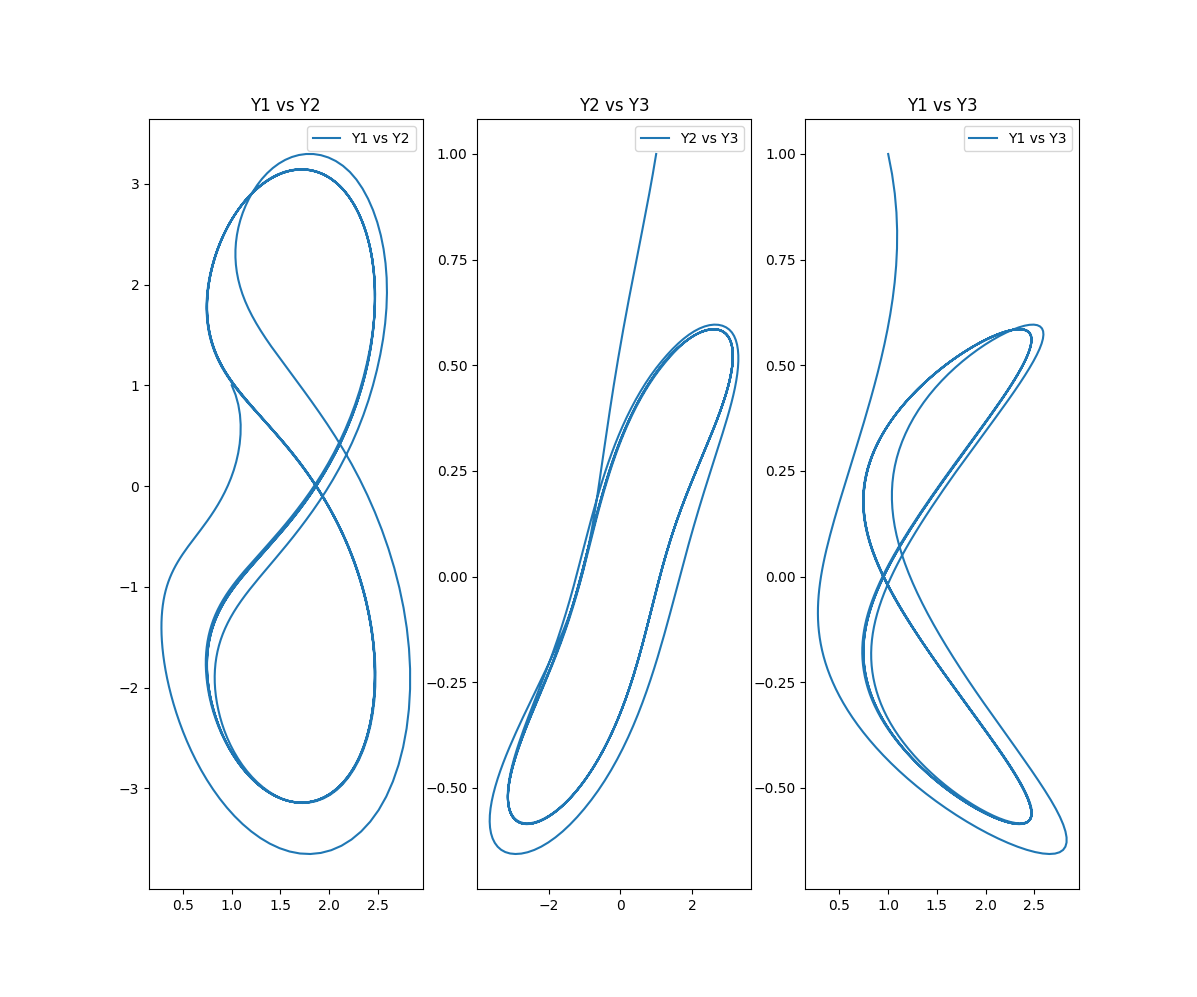
:

Приблизим и получим:



Полученные результаты подтверждают, что начальные значения оказывают влияние лишь на начальную амплитуду колебаний , не изменяя при этом их устойчивое поведение в долгосрочной перспективе. Возможности применения данной системы будут подробно рассмотрены в разделе выводов.

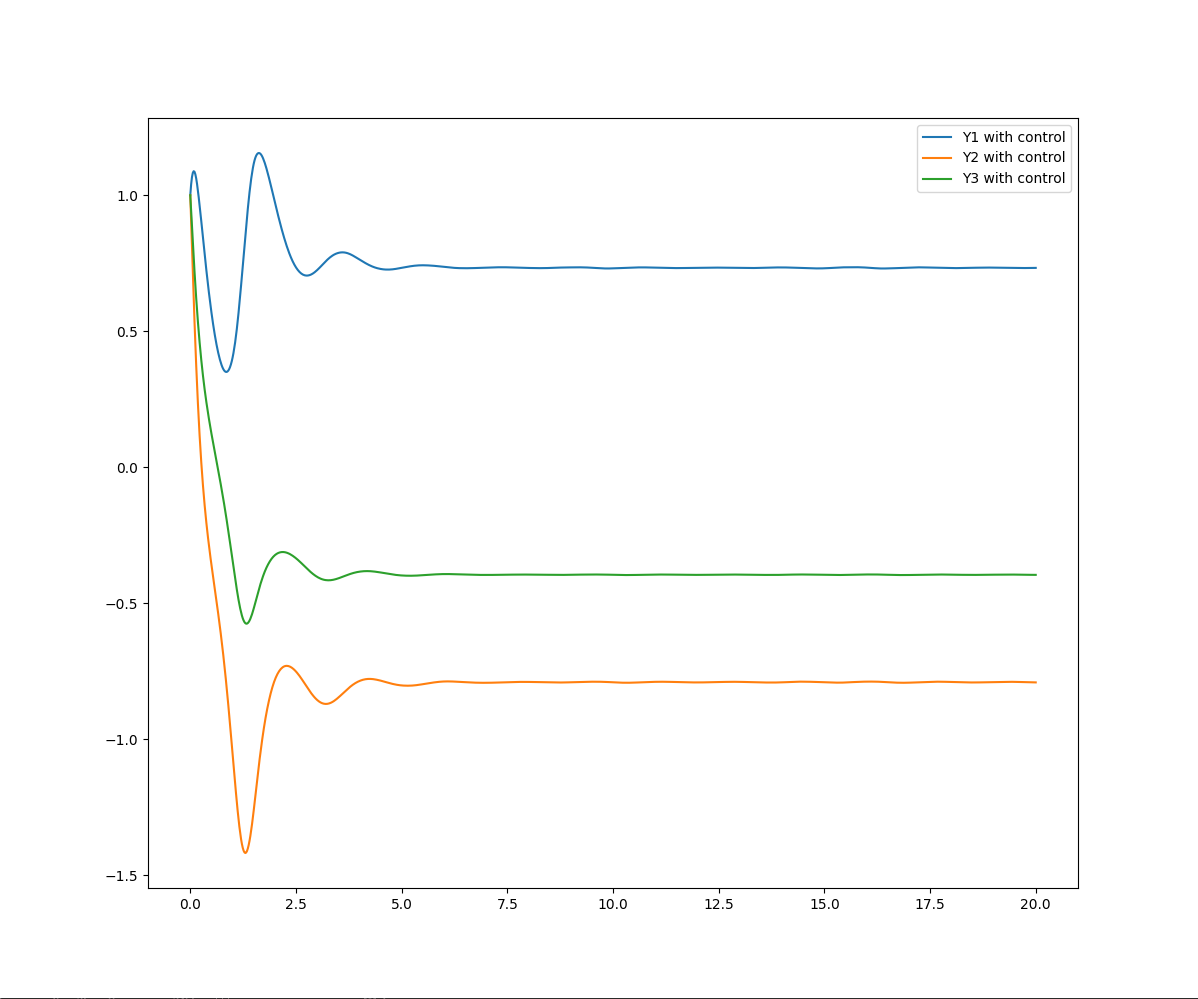
Проведем анализ фазовых портретов при :



Каждый из представленных фазовых портретов демонстрирует наличие периодических колебаний или предельного цикла, что указывает на способность системы стабилизироваться в определенном повторяющемся состоянии. Кроме того, все портреты свидетельствуют о выраженных взаимодействиях между переменными, что подтверждается характеристиками исследуемого объекта

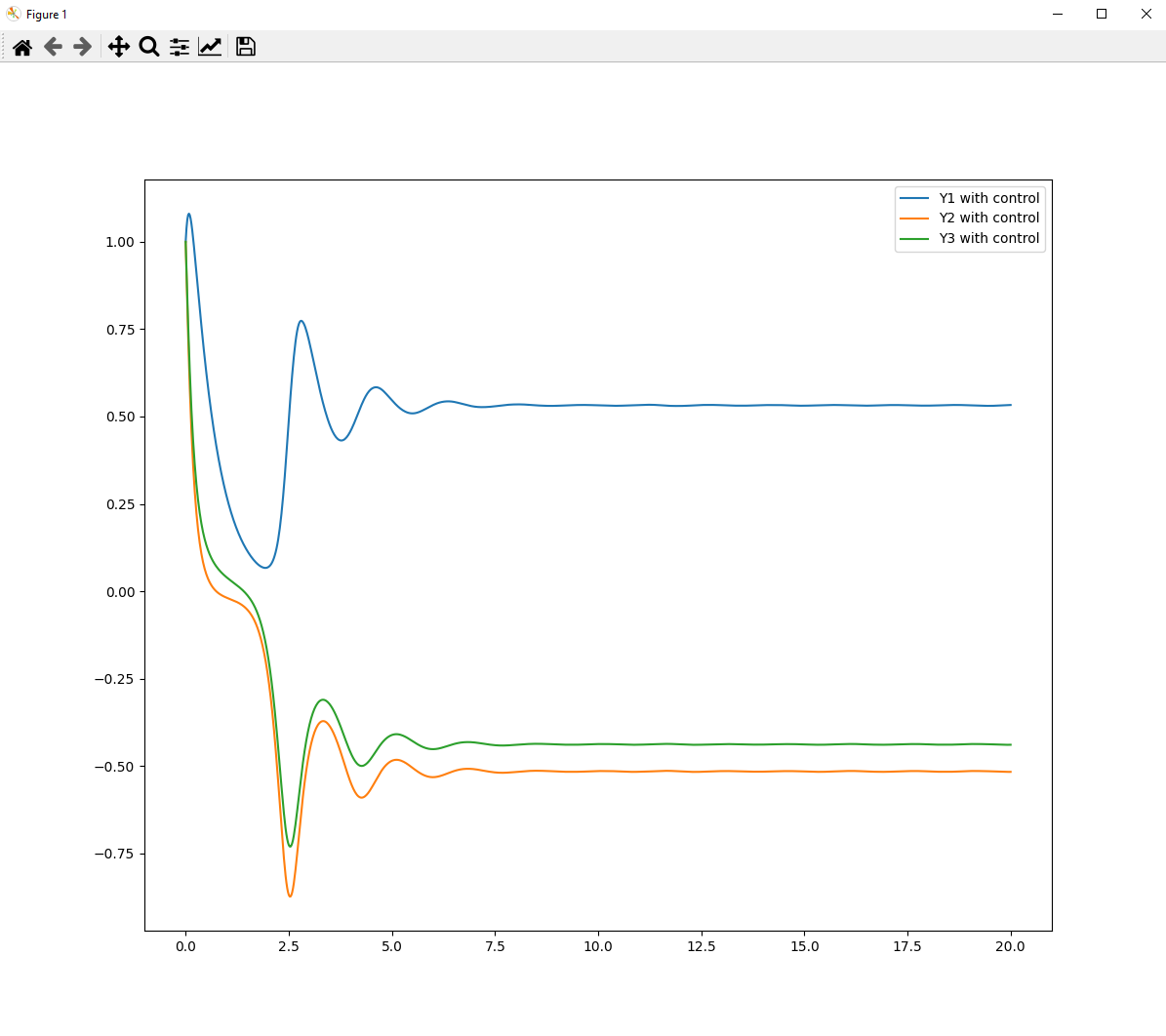
Проанализируем, как параметр ρ влияет на функционирование системы. Для этого зададим значение T1 равным 1, чтобы исключить его влияние на общую динамику системы.

При ρ = 0.5:

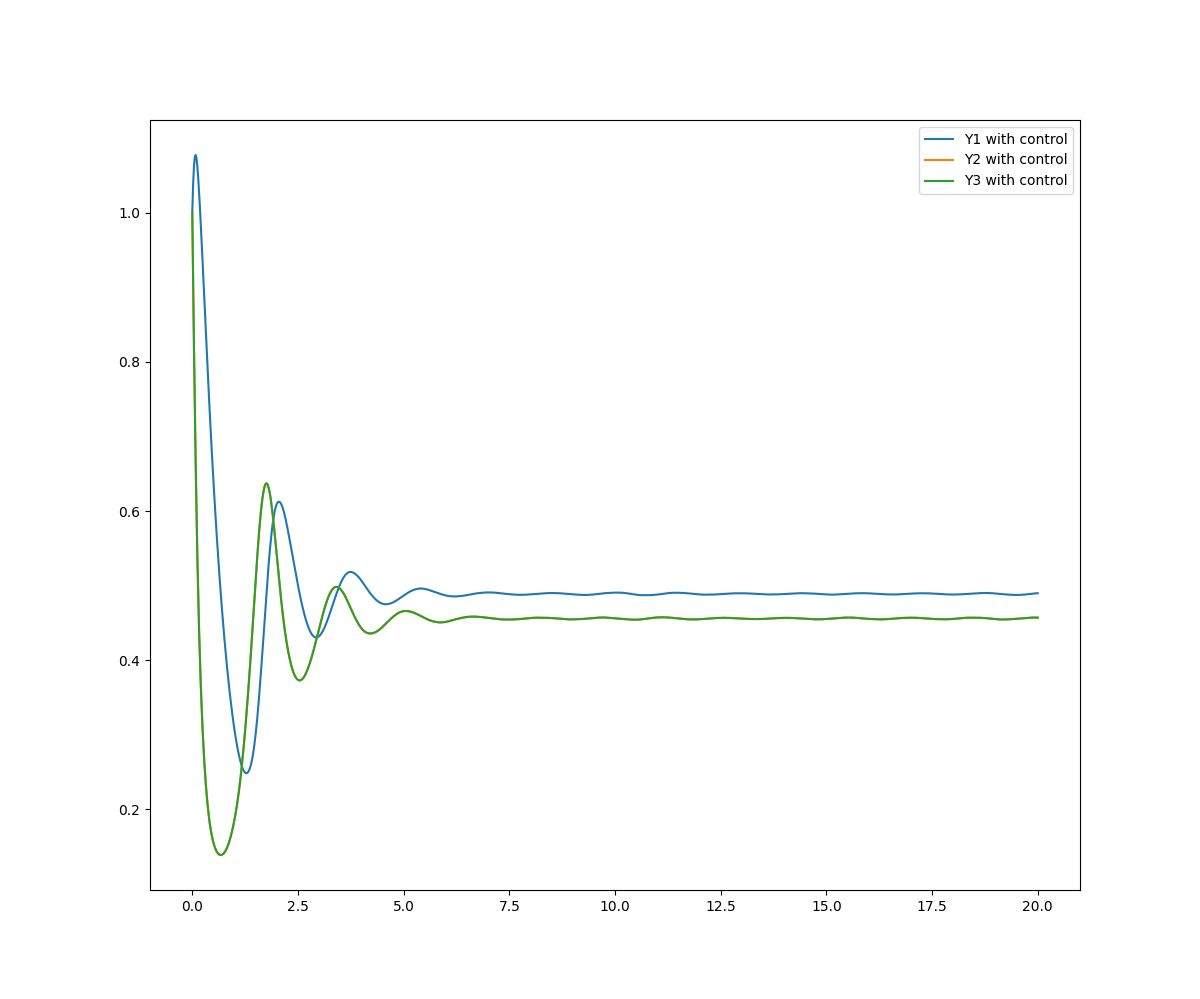


На графике показано поведение трёх переменных Y1​, Y2, Y3​ во времени при наличии управления. На начальном этапе (0–2 секунды) все переменные демонстрируют затухающие осцилляции: Y1 стабилизируется на уровне чуть выше 1, Y2 – около -1, а Y3​ – около -0.5. Колебания всех переменных имеют убывающую амплитуду, что свидетельствует о частичной устойчивости системы. После 5 секунд переменные выходят на стационарное поведение, полностью прекращая осцилляции. Управление эффективно стабилизирует систему, сводя динамику к постоянным значениям.

Рассмотрим другие значия, например ρ = 0.85:

**

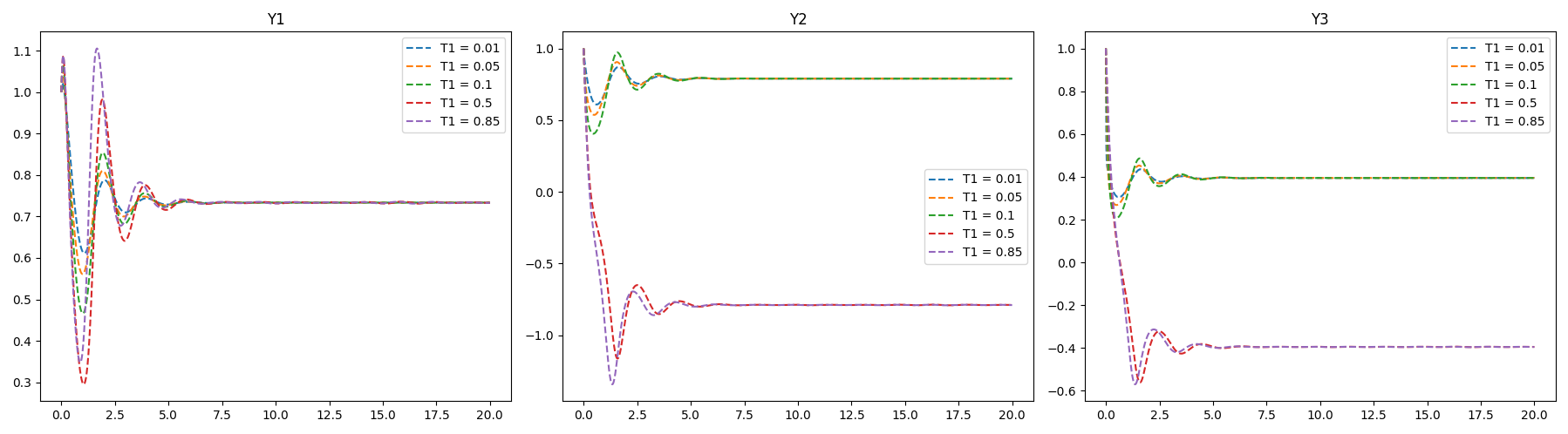
При ρ = 1:



После периода переходных процессов все три переменные стабилизируются около определённых значений. Это подтверждается как аналитически, так и графически.

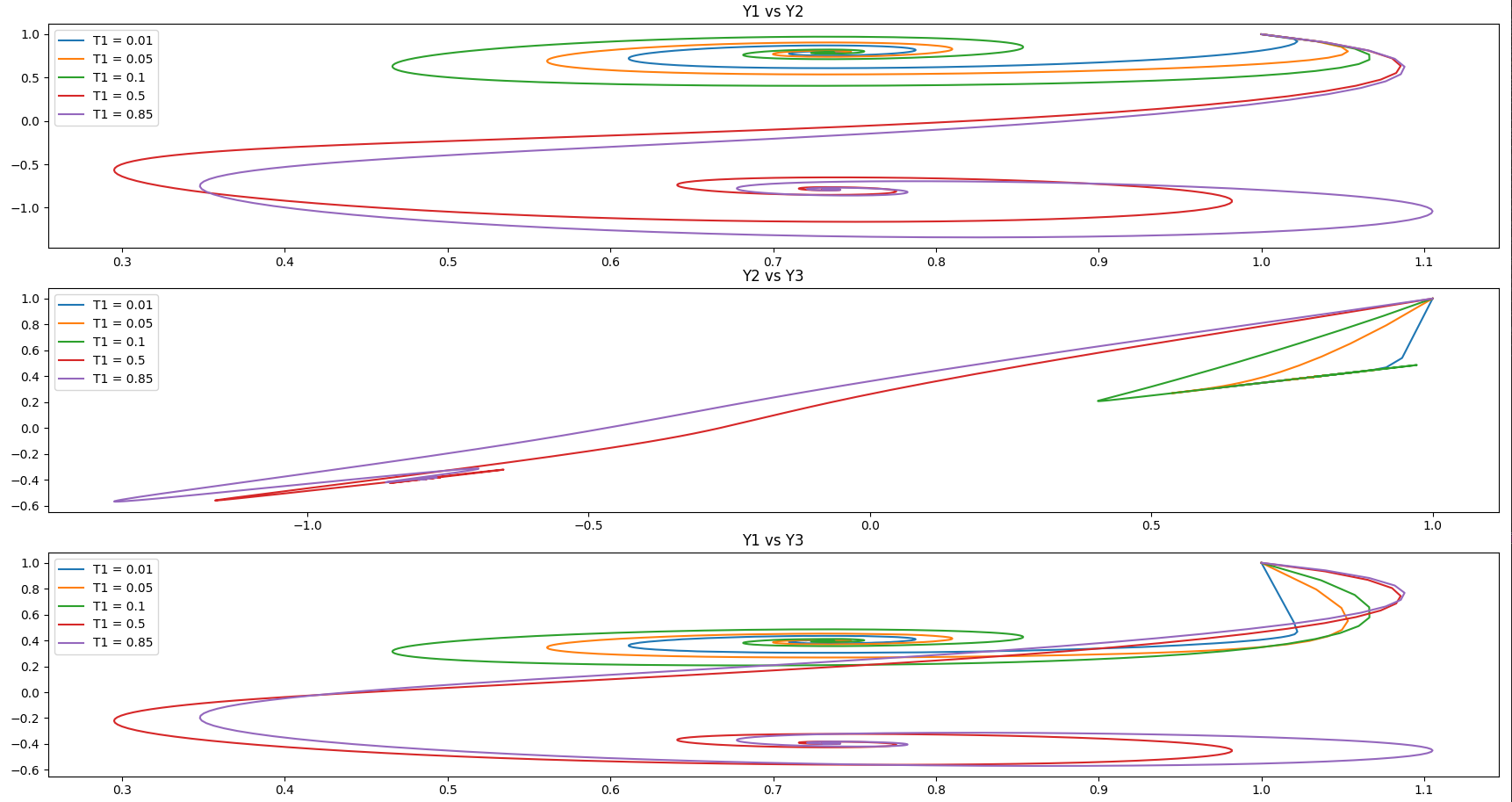
Установим в начальное значение и проанализируем влияние T1:

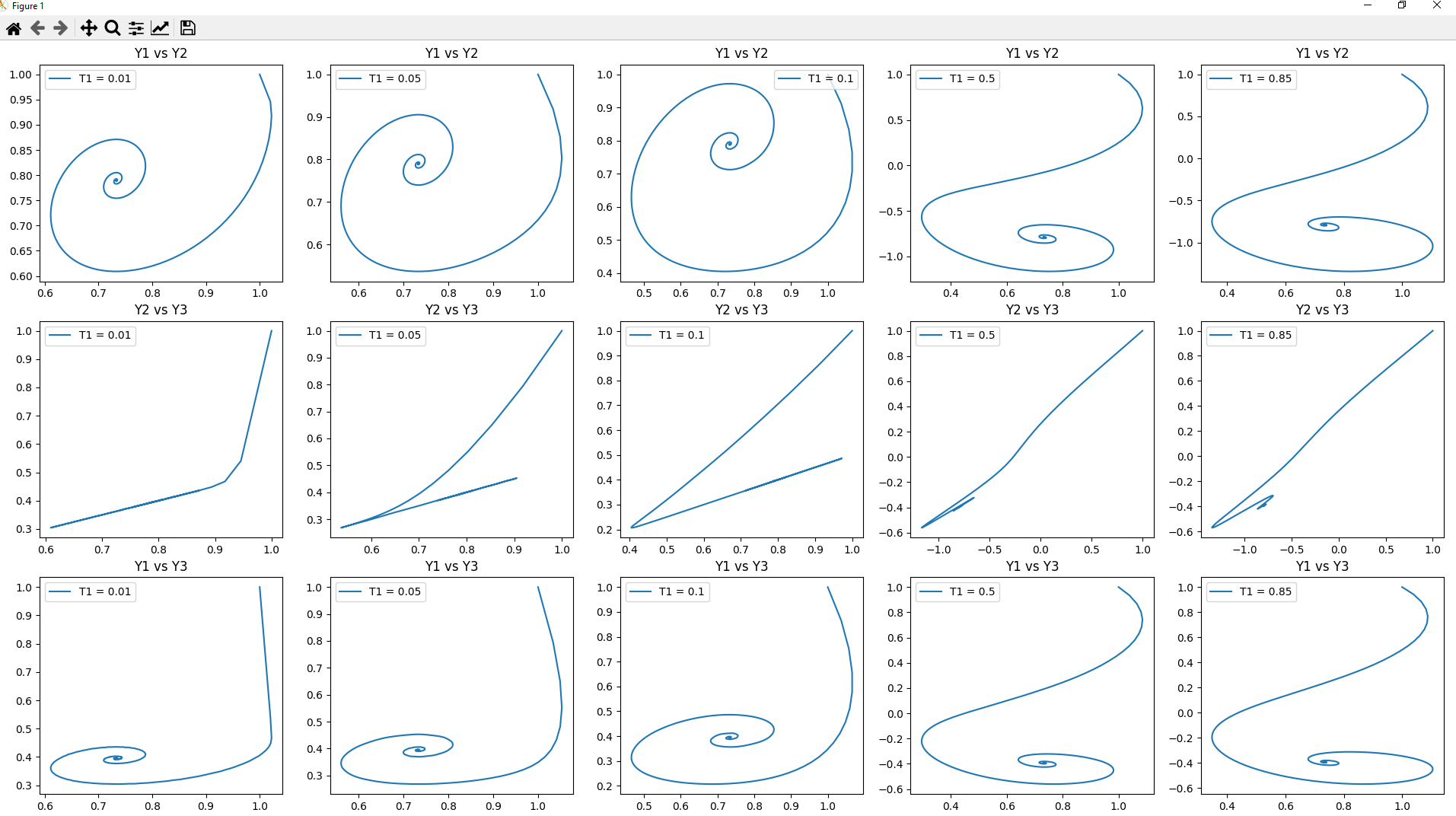
Рассмотрим случаи, когда T1 = 0.01, 0.05, 0.1, 0.5, 0.85:



С уменьшением параметра T1​ система стабилизируется быстрее за счёт увеличения интенсивности управляющего воздействия, которое способствует ускорению перехода к устойчивому состоянию. Наоборот, при более высоких значениях T1​ процесс стабилизации замедляется, поскольку управление становится менее интенсивным, что продлевает время переходного периода.

Рассмотрим и проанализируем фазовые портреты:





Исходя из полученных результатов можно сделать вывод, что при меньших T1 система демонстрирует более быструю стабилизацию. Напротив, при более высоких значениях T1 фазовые портреты становятся более сглаженными, что указывает на более низкую интенсивность коррекционного управления и более мягкое воздействие на систему.

Проанализируем график макропеременной и график управления:

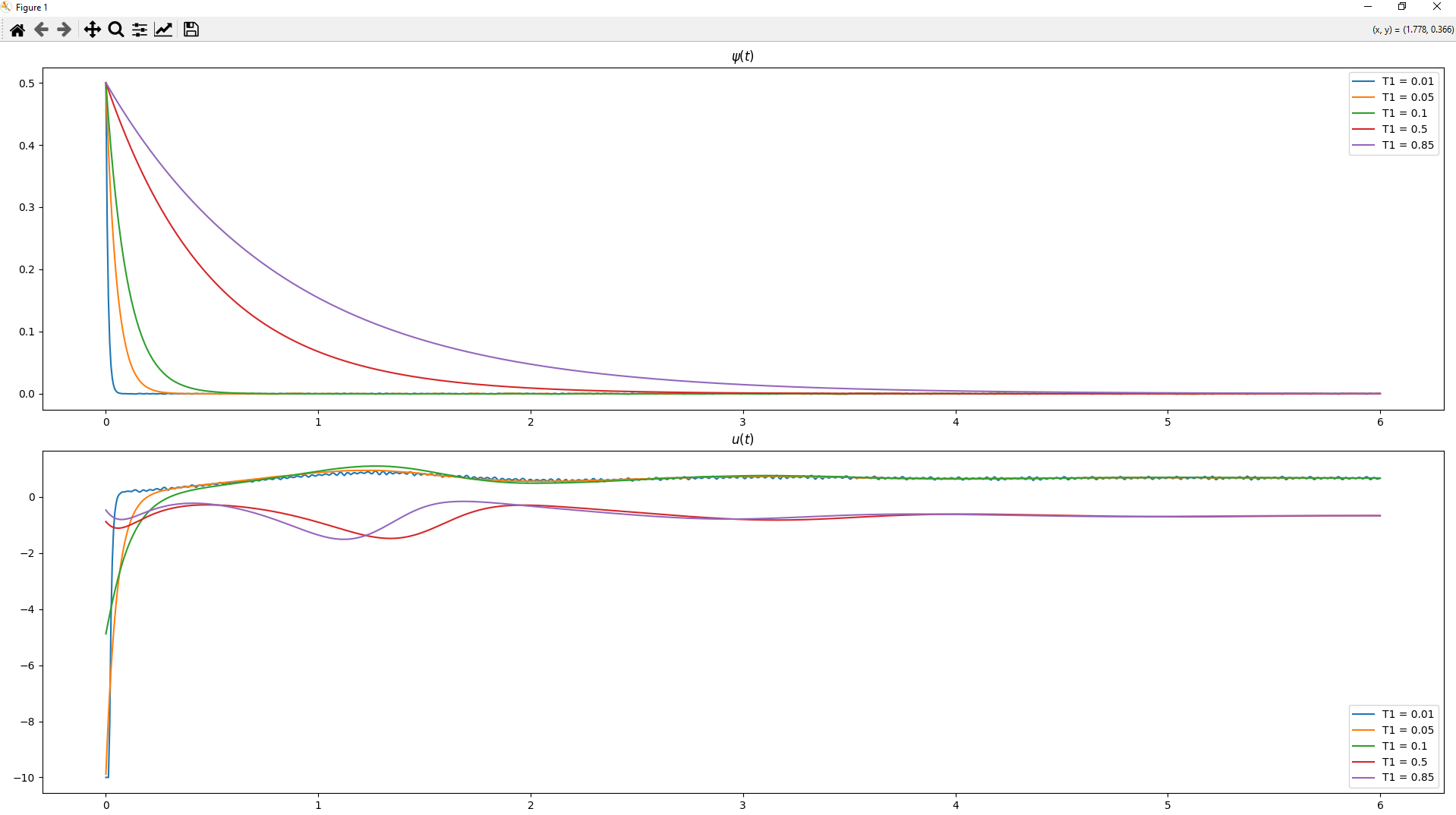
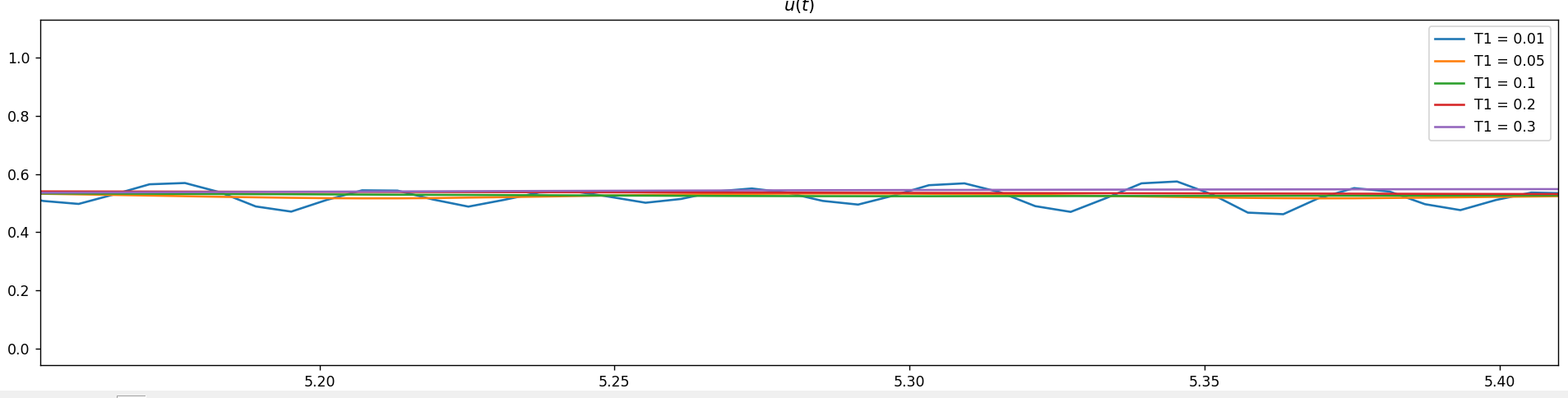


График макропеременной также подверждает, что при более низких T1 происходит резкий переход в точку стабилизации, а при более высоких переход происходит плавно. Стабилизация функции управления при более низких T1 по поведению похожа на поведение макропеременной – резкий переход в точку стабилизации. Однако, при приближении можно заметить, что функция управления колеблется вокруг точки стабилизации, так как интенсивное коррекционное управление не дает ей стабилизироваться в этой точке. Функция как бы «перескакивает» свою точку стабилизации с разных сторон.

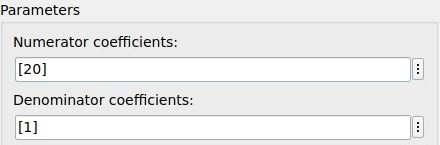
При более высоких T1 поведение предсказуемо – более плавная, но и более медленная стабилизация.



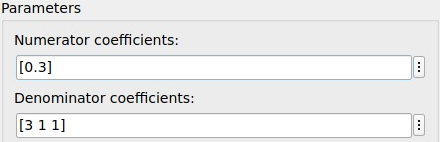
**Выполнение части 2:**

Для моделирования передаточной функции воспользуемся блоком Transfer FCN. Его параметры:

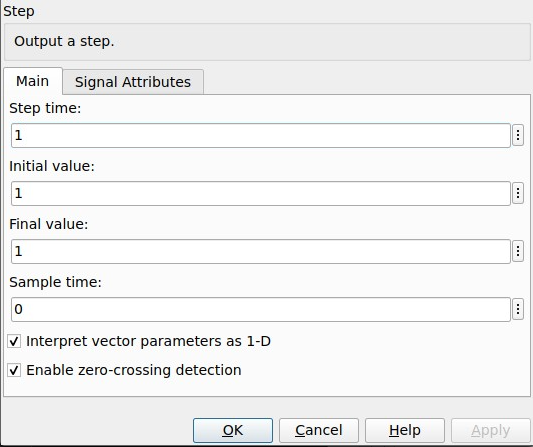
Для W1 (по заданию это константа = 20):



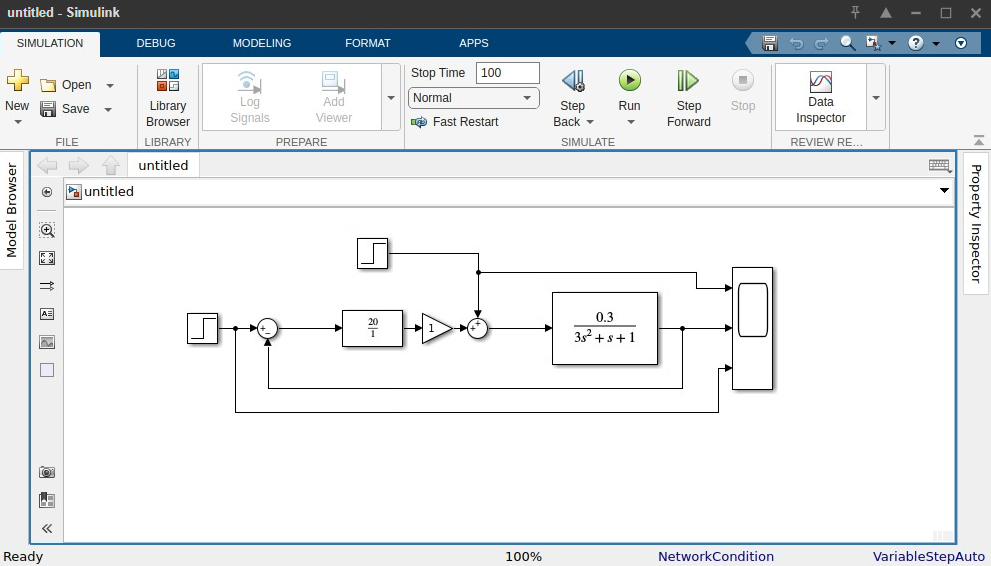
Для W2 (по заданию это ):



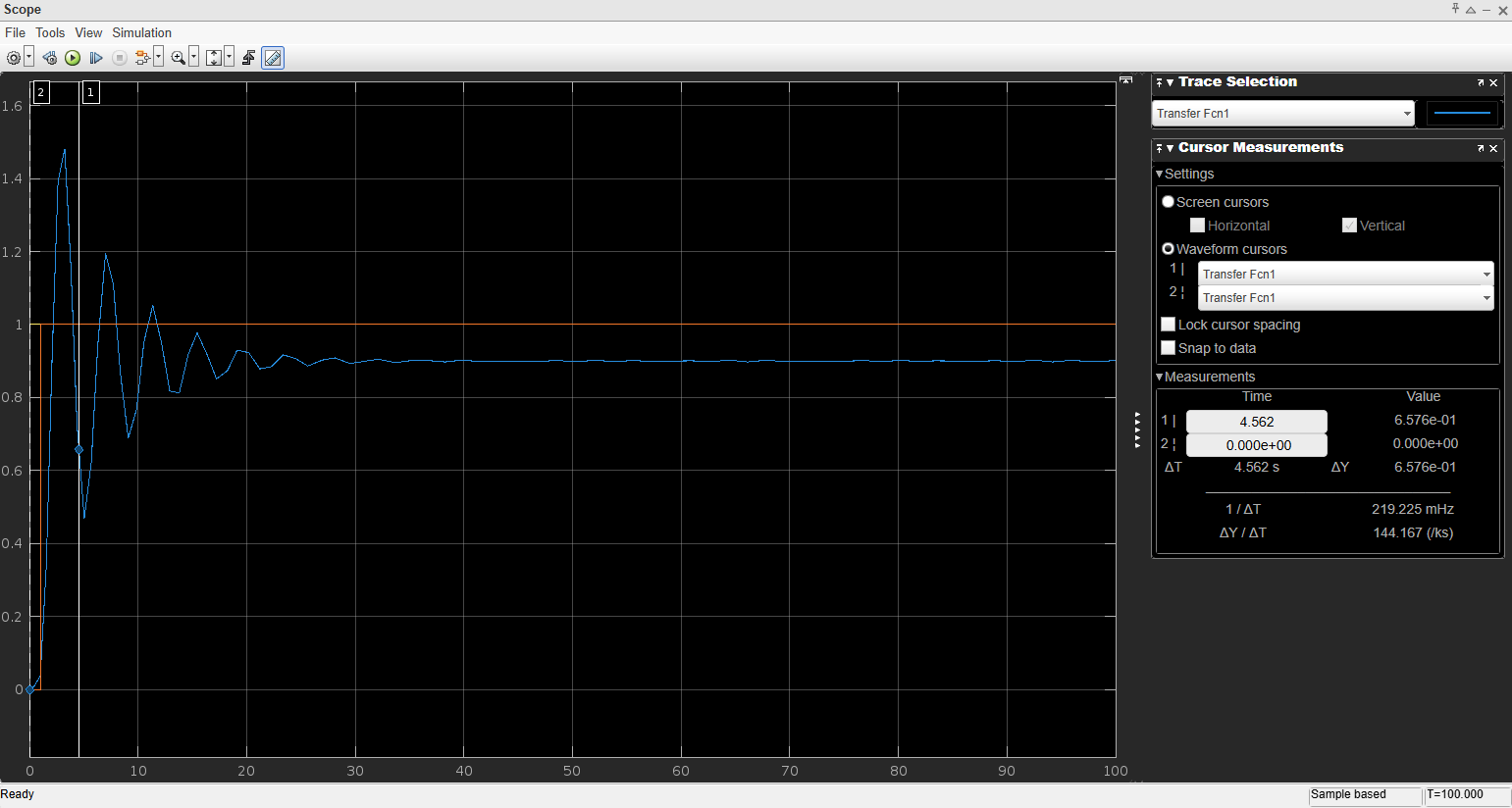
Возмущение установлено в 0:



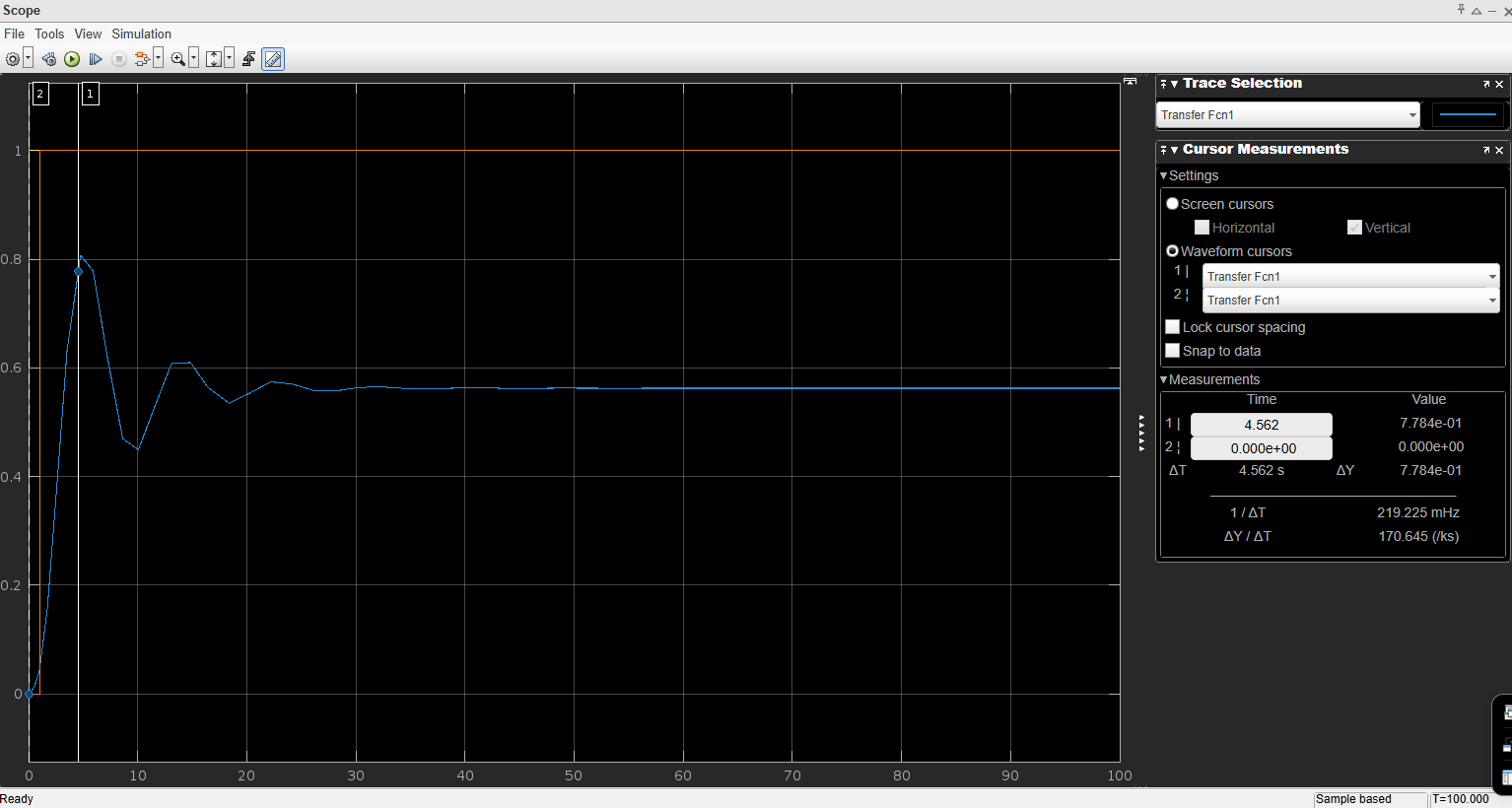
Общий вид схемы:



**Результат scope:**

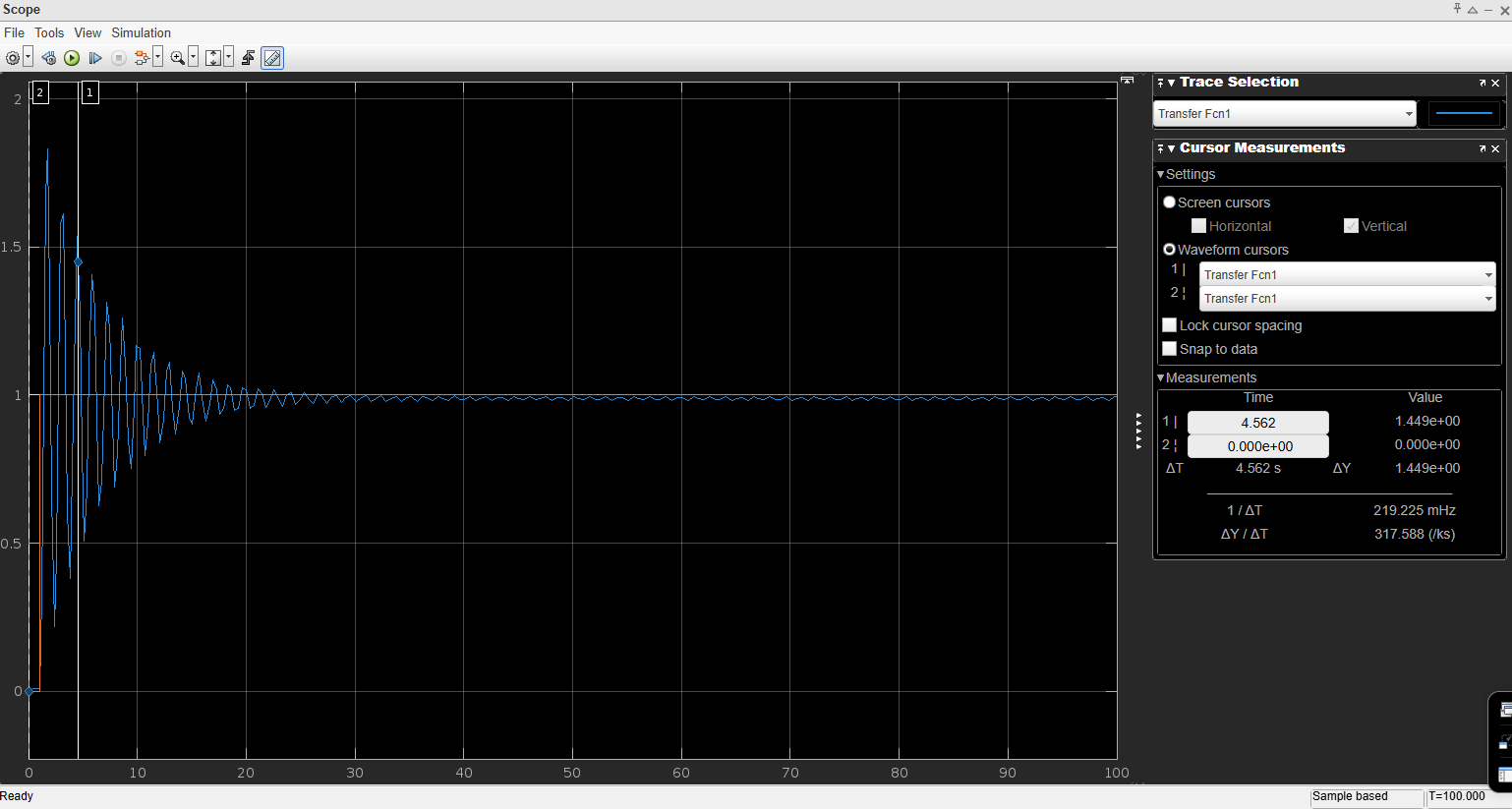


**Результат работы с предделителем для W1 = 0.1**

****

**Общие выводы по модели**: низкая степени перерегулирования, крайне высокая степень затухания, практически двухкратное значение декремента колебаний. Модель показывает себя хорошо, степень затухания чуть выше нормы в 0.95

**Результат работы с множителем W1 = 10:**

****

**Общие выводы по модели**: самая высокая степень перерегулирования, самая низкая степень затухания, как и логарифмический декремент колебаний. Модель показала себя хуже остальных.

1. **Выводы**

**Часть 1:**

Модель Лоренца, рассмотренная в первой части лабораторной работы, нашла применение в различных областях, хотя изначально была создана для описания динамики атмосферных процессов. Благодаря своей высокой чувствительности к начальным условиям, она стала одной из первых моделей, демонстрирующих проявление хаоса в реальных климатических системах. В контексте теории хаоса модель Лоренца является классическим примером детерминированного хаоса, когда даже незначительные изменения начальных параметров приводят к значительным различиям в поведении системы, несмотря на строгую зависимость от заданных уравнений.

Модель, рассмотренная в первой части без учета управления, находит применение в различных областях благодаря своей периодической природе. Она может быть полезна для анализа колебательных и электронных систем (таких как осцилляторы), а также для изучения основ теории управления, что было продемонстрировано на практике.

В результате анализа модели с управлением, можно выделить следующие определения:

* Переменная ρ регулирует степень корректировки для стабилизации системы, иными словами, показывает, насколько значима Y2 в контексте Y3.
* Макропеременная необходима для обратной связи в функции управления. Функция управления будет корректировать определяющие функции модели до тех пор, пока макропеременная не станет равна нулю.
* Сама функция управления так же, как и определяющие функции системы, с течением времени доходит до своей точки стабилизации.

Влияние параметра T1:

* В общем случае, параметр T1 влияет на ключевые параметры системы схожим образом – более низкие его значения дают усиленную коррекцию, что приводит к более активной реакции системы на изменение макропеременной. Однако, слишком малое его значение может привести к высокой степени перерегулирования и колебаниям.
* В то же время, более высокие его значения дают ослабленную коррекцию, то есть реакция системы становится более плавной, но в то же время скорость стабилизации будет ниже.
* Данное поведение подтвердилось на всех исследуемых свойствах модели – значения Y1, Y2, Y3, соответствующие фазовые портреты, макропеременная, и функция управления. В функции управления была зафиксирована и отрицательная сторона низких значений T1 – значение функции достаточно сильно колебалось возле точки стабильности относительно более высоких значений.

Система в любом случае достигает устойчивого состояния, выбор T1 влияет на скорость достижения этого состояния и характер траектории схождения.

**Часть 2:**

С помощью ПО Matlab получилось построить схему модели и проанализировать её результативность.

По результатам анализа моделей наилучшей оказалась модель с W1 = 10.

**Листинг программы**

|  |
| --- |
| import numpy as np  import matplotlib.pyplot as plt  from scipy.integrate import solve\_ivp  alpha\_val = 4  beta\_val = 9  gamma\_val = 1.7  mu\_val = 2.2  lambda\_val = 3.12  delta\_val = 0.7  rho\_val = 0.5  T1\_val = [1]  time\_interval = 20  time\_points = np.linspace(0, time\_interval, 1000)  initial\_conditions = [1, 1, 1]  def calc\_psi(v2, v3):      return v3 - rho\_val \* v2  def calc\_u(v1, v2, v3, T1\_val):      return lambda\_val \* v3 - delta\_val \* v2 + rho\_val \* (mu\_val \* (v2 + v3) - beta\_val \* v1 \* v3) - calc\_psi(v2, v3) / T1\_val  def no\_control\_system(t, vars):      v1, v2, v3 = vars      dv1\_dt = alpha\_val \* v2 \* v3 - gamma\_val \* v1      dv2\_dt = mu\_val \* (v2 + v3) - beta\_val \* v1 \* v3      dv3\_dt = delta\_val \* v2 - lambda\_val \* v3      return [dv1\_dt, dv2\_dt, dv3\_dt]  def extract\_macro\_and\_control(t, vars):      v1, v2, v3 = vars      psi\_val = calc\_psi(v2, v3)      u\_val = calc\_u(v1, v2, v3, T1\_val)      return psi\_val, u\_val  def controlled\_system(t, vars, T1\_val):      v1, v2, v3 = vars      control\_val = calc\_u(v1, v2, v3, T1\_val)      dv1\_dt = alpha\_val \* v2 \* v3 - gamma\_val \* v1      dv2\_dt = mu\_val \* (v2 + v3) - beta\_val \* v1 \* v3      dv3\_dt = delta\_val \* v2 - lambda\_val \* v3 + control\_val      return [dv1\_dt, dv2\_dt, dv3\_dt]  sol\_no\_control = solve\_ivp(no\_control\_system, [0, time\_interval], initial\_conditions, t\_eval=time\_points)  sol\_with\_control = solve\_ivp(controlled\_system, [0, time\_interval], initial\_conditions, t\_eval=time\_points, args=T1\_val)  Y1\_no\_control, Y2\_no\_control, Y3\_no\_control = sol\_no\_control.y  Y1\_with\_control, Y2\_with\_control, Y3\_with\_control = sol\_with\_control.y  psi\_with\_control\_vals = [extract\_macro\_and\_control(ti, [Y1\_with\_control[i], Y2\_with\_control[i], Y3\_with\_control[i]])[0] for i, ti in enumerate(time\_points)]  u\_with\_control\_vals = [extract\_macro\_and\_control(ti, [Y1\_with\_control[i], Y2\_with\_control[i], Y3\_with\_control[i]])[1] for i, ti in enumerate(time\_points)]  plt.figure(figsize=(12, 10))  plt.plot(time\_points, Y1\_no\_control, label='Y1')  plt.plot(time\_points, Y2\_no\_control, label='Y2')  plt.plot(time\_points, Y3\_no\_control, label='Y3')  plt.legend()  plt.show()  plt.figure(figsize=(12, 10))  plt.subplot(1, 3, 1)  plt.plot(Y1\_no\_control, Y2\_no\_control, label='Y1 vs Y2')  plt.title('Y1 vs Y2')  plt.legend()  plt.subplot(1, 3, 2)  plt.plot(Y2\_no\_control, Y3\_no\_control, label='Y2 vs Y3')  plt.legend()  plt.title('Y2 vs Y3')  plt.subplot(1, 3, 3)  plt.plot(Y1\_no\_control, Y3\_no\_control, label='Y1 vs Y3')  plt.title('Y1 vs Y3')  plt.legend()  plt.show()  plt.figure(figsize=(12, 10))  plt.plot(time\_points, Y1\_with\_control, label='Y1 with control')  plt.plot(time\_points, Y2\_with\_control, label='Y2 with control')  plt.plot(time\_points, Y3\_with\_control, label='Y3 with control')  plt.legend()  plt.show()  T1\_values = [0.01, 0.05, 0.1, 0.5, 0.85]  results = {T1: None for T1 in T1\_values}  for T1 in T1\_values:      sol = solve\_ivp(          controlled\_system,          [0, time\_interval],          initial\_conditions,          t\_eval=time\_points,          args=(T1,)      )      results[T1] = sol.y  fig, axes = plt.subplots(1, 3, figsize=(18, 5))  # 3 подграфика в строку  titles = ['Y1', 'Y2', 'Y3']  for i, ax in enumerate(axes):      for T1 in T1\_values:          ax.plot(              time\_points,              results[T1][i],              label=f'T1 = {T1}',              linestyle='--'          )      ax.set\_title(titles[i])      ax.legend()  plt.tight\_layout()  plt.show()  plt.figure(figsize=(18, 12))  plt.subplot(3, 1, 1)  for T1 in T1\_values:      sol = solve\_ivp(controlled\_system, [0, time\_interval], initial\_conditions,                      t\_eval=time\_points, args=(T1,))      Y1, Y2, Y3 = sol.y      plt.plot(Y1, Y2, label=f"T1 = {T1}")  plt.title("Y1 vs Y2")  plt.legend()  plt.subplot(3, 1, 2)  for T1 in T1\_values:      sol = solve\_ivp(controlled\_system, [0, time\_interval], initial\_conditions,                      t\_eval=time\_points, args=(T1,))      Y1, Y2, Y3 = sol.y      plt.plot(Y2, Y3, label=f"T1 = {T1}")  plt.title("Y2 vs Y3")  plt.legend()  plt.subplot(3, 1, 3)  for T1 in T1\_values:      sol = solve\_ivp(controlled\_system, [0, time\_interval], initial\_conditions,                      t\_eval=time\_points, args=(T1,))      Y1, Y2, Y3 = sol.y      plt.plot(Y1, Y3, label=f"T1 = {T1}")  plt.title("Y1 vs Y3")  plt.legend()  plt.tight\_layout()  plt.show()  plt.figure(figsize=(18, 12))  for idx, T1 in enumerate(T1\_values):      sol = solve\_ivp(controlled\_system, [0, time\_interval], initial\_conditions,                      t\_eval=time\_points, args=(T1,))      Y1, Y2, Y3 = sol.y      plt.subplot(3, len(T1\_values), idx + 1)      plt.plot(Y1, Y2, label=f"T1 = {T1}")      plt.title(f"Y1 vs Y2")      plt.legend()      plt.subplot(3, len(T1\_values), idx + len(T1\_values) + 1)      plt.plot(Y2, Y3, label=f"T1 = {T1}")      plt.title(f"Y2 vs Y3")      plt.legend()      plt.subplot(3, len(T1\_values), idx + 2 \* len(T1\_values) + 1)      plt.plot(Y1, Y3, label=f"T1 = {T1}")      plt.title(f"Y1 vs Y3")      plt.legend()  plt.tight\_layout()  plt.show()  plt.figure(figsize=(18, 12))  psi\_vals\_dict = {}  u\_vals\_dict = {}  time\_interval = 6  time\_points = np.linspace(0, time\_interval, 1000)  for T1\_val in T1\_values:      sol = solve\_ivp(controlled\_system, [0, time\_interval], initial\_conditions, t\_eval=time\_points, args=(T1\_val,))      Y1, Y2, Y3 = sol.y      t\_vals = sol.t      psi\_vals = []      u\_vals = []      for t, vars in zip(t\_vals, zip(Y1, Y2, Y3)):          psi, u = extract\_macro\_and\_control(t, vars)          psi\_vals.append(psi)          u\_vals.append(u)      u\_vals\_clipped = np.clip(u\_vals, -10, 20)      psi\_vals\_dict[T1\_val] = psi\_vals      u\_vals\_dict[T1\_val] = u\_vals\_clipped  plt.subplot(2, 1, 1)  for T1\_val in T1\_values:      plt.plot(t\_vals, psi\_vals\_dict[T1\_val], label=f"T1 = {T1\_val}")  plt.title("$\psi(t)$")  plt.legend()  plt.subplot(2, 1, 2)  for T1\_val in T1\_values:      plt.plot(t\_vals, u\_vals\_dict[T1\_val], label=f"T1 = {T1\_val}")  plt.title("$u(t)$")  plt.legend()  plt.tight\_layout()  plt.show() |