МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики

Кафедра информационных технологий

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4
по дисциплине
«МНОГОАГЕНТНОЕ МОДЕЛИРОВАНИЕ»

Выполнил студент группы 45/2 ___________________ Т. Э. Айрапетов

Направление подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем
Курс 4

Отчет принял доктор физико-математических наук,
профессор ____ А.И. Миков

Краснодар
2024 г.

Задание: провести моделирование функционирования динамической среды с агентом во времени, исследовать систему на устойчивость. Динамическая среда (объект управления) описывается обыкновенным дифференциальным уравнением 2-го порядка с постоянными положительными коэффициентами. Датчики (устройства сборка информации в среде) имеют запаздывание величины τ, в результате чего агент (регулятор) действует на основе устаревшей информации.

При τ = 0 система устойчива. При небольшом увеличении τ устойчивость сохраняется, но при дальнейшем увеличении τ устойчивость может потеряться. Определить зависимости величины граничного значения запаздывания τ, при которых система ещё устойчива, от других числовых параметров (n, m, a0, b0, a1, b1, ...) среды и агента (k). Функция цели g(t) = H(t) - функция Хевисайда.

Рекомендации

Время в задаче непрерывно, но для расчетов на компьютере нужно провести дисркетизацию задачи и решения. Обозначим Δt - шаг по времени, например 0.01. Тогда состояния динамической среды изменяются в дискретные моменты времени $t_i = 0, \Delta t, 2\Delta t, 3\Delta t, \ldots$ ($t_i = i\Delta t$). Состояния среды в эти моменты обозначим $x_i = x(t_i)$, значения управления $u_i = u(t_i)$. Начальное состояние среды (0,0,0, ..., 0). Дифференциальное уравнение заменяется разностным.

Моделирование начинается в момент $t = 0$ и заканчивается в момент $t{=}t_{stop}$, когда станет очевидно, что процесс сходится (устойчивость) или что процесс расходится (неустойчивость).

Решение.

Для решения задачи моделирования, для уравнения (1) была рассчитана разностная схема (2) и взят шаг дискретизации равный 0.01.

Общий вид уравнения:

$$ax'' + bx' + cx = u(t) \; (1)$$

Общий вид схемы:

$$x_{n+1} = \frac{2ax_n - \left(a - \frac{b\Delta t}{2}\right)x_{n-1} + \Delta t^2 (u_n - c \cdot x_n)}{a + \frac{b\Delta t}{2}} \quad (2)$$

Далее в коде были описаны шаги итеративного расчета *x* до некоторого условного понятия сходимости. Будем считать, что система сходится, если к концу вычислений стандартное отклонение последних 10% *x* не больше epsilon=1e-6. Также введем значение максимального количества шагов (1000) для случаев, когда система заведомо расходится.

Проведены эксперименты для значений *a*, *b*, *c*, *k* из списка [0.5, 1, 2, 5, 10] и для τ в диапазоне от 2 до 5 единиц отставания.

Текст программы на языке Python:

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd



T = 1000 # Время моделирования
dt = 0.01  # Шаг по времени
N_max = int(T / dt)  # Максимальное количество шагов
N_min = 10 # Минимальное количество шагов

epsilon = 1e-6  # Порог сходимости

data = []

def simulate(a, b, c, k, tau):

    x = [0, 0]
    g = lambda t: 1 if t > 0 else 0

    def get_x(t):
        if t <= 0:
            return x[0]
        return x[t+1]
```

```
    u = lambda t: k*(g(dt*(t-tau))-get_x(t-tau))

    t = 1

    while (abs(np.std(x[-int(len(x)*0.2):])) > epsilon or t < max(tau+2, N_min)) and t < N_max - 1:

      x.append((2 * a * x[-1] - (a - b * dt / 2) * x[-2] + dt**2 * (u(t) - c * x[-1])) / (a + b * dt / 2))
      t += 1

    return x, t


  temp = [0.5, 1, 2, 5, 10]
  for a in temp:
    for b in temp:
      for c in temp:
        for k in temp:
          for tau in range(2, 5):
            x, t = simulate(a,b,c,k,tau)
            data.append((a,b,c,k,tau,np.std(x[-int(len(x)*0.1):]), x[-1], t*dt/T))
```

По итогам моделирования различных ситуаций можно сделать некоторые выводы о зависимости между значениями параметров и сходимостью системы. На рисунке 1 можно увидеть heatmap по корреляции наших параметров.

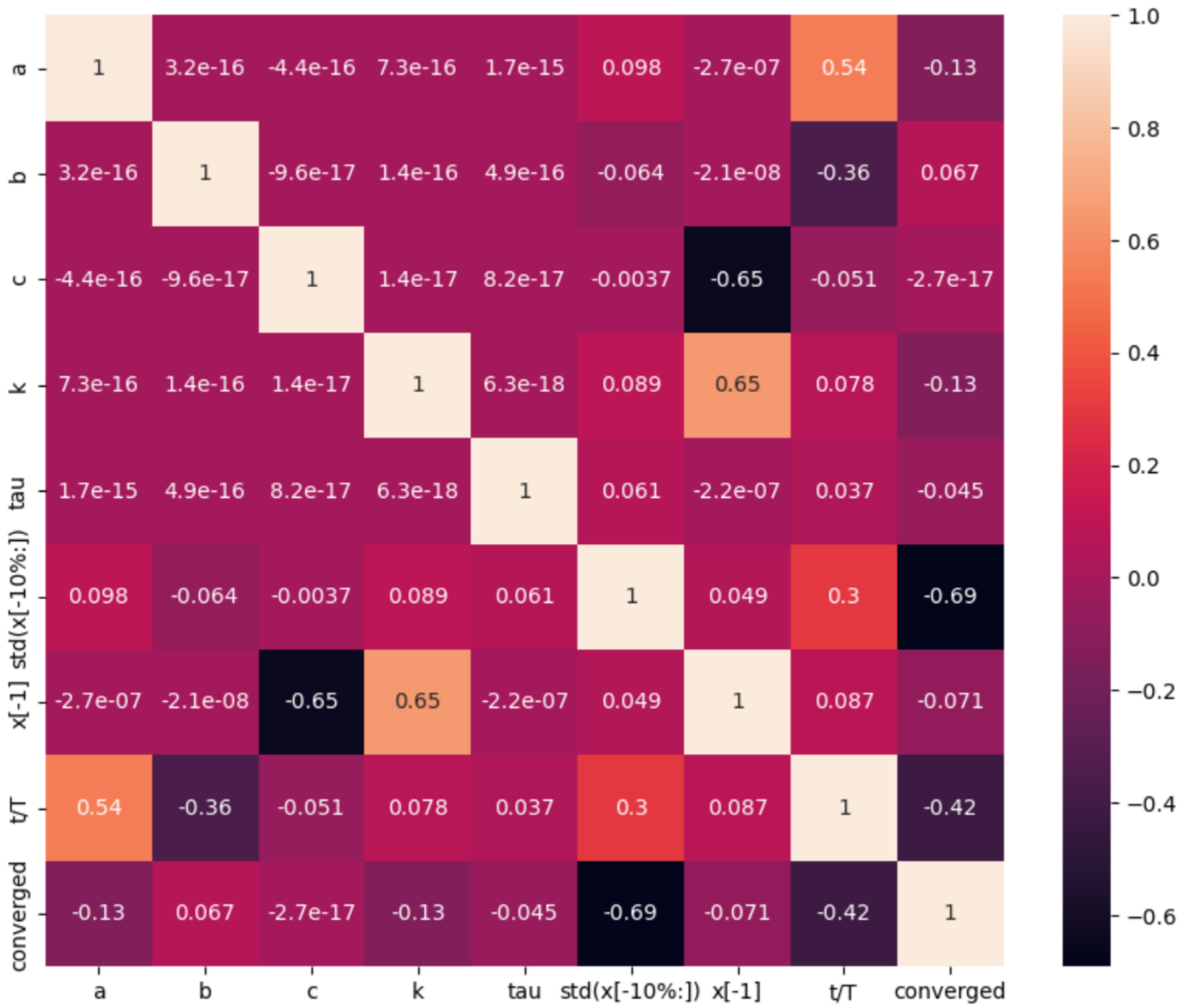

Рисунок 1 – heatmap корреляции параметров

Из рисунка 1 можно увидеть некоторые зависимости между параметрами. Например, параметр converged (булевый признак, отвечающий на вопрос, сошлась система до лимита по шагам или нет) связан с ростом значения параметра *b,* т. е. при больших значениях этого параметра система скорее всего сойдется. Пример графика с увеличенным значением параметра *b* можно увидеть на рисунке 2.

Параметр *a*, в свою очередь, влияет на скорость схождения в обратную сторону, т. е. с его ростом время схождения увеличивается. Пример влияния параметра *a* на кривую сходимости можно увидеть на рисунке 3.

Параметр *c* также негативно влияет на сходимость - в меньшей мере на скорость схождения, в большей на конечное значение системы.

Параметр *k*, в какой-то мере уравновешивает параметр *c*, так как при его правильном задании, система сойдется до нужного значения (1). Тем не менее, слишком большие значения могут привести к увеличению времени схождения.

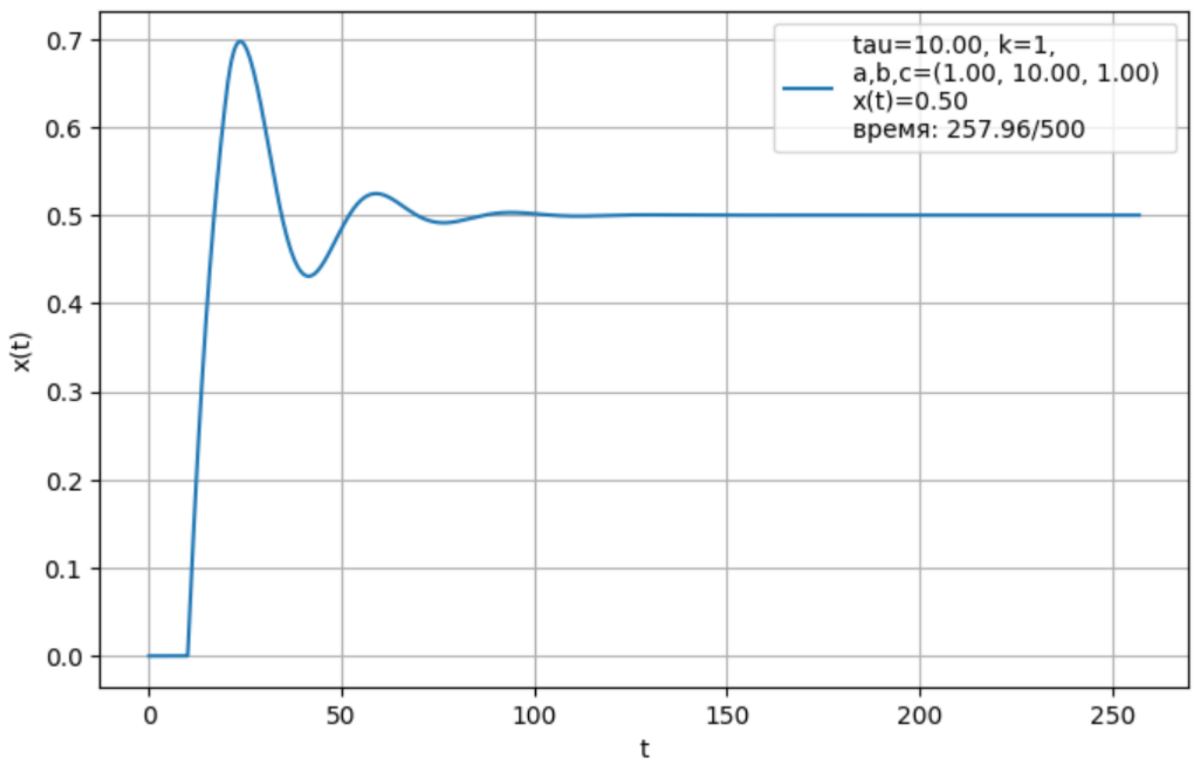

Рисунок 2 - График состояния системы при большой задержке и большим b. Можно сказать, что b нивелирует задержку.

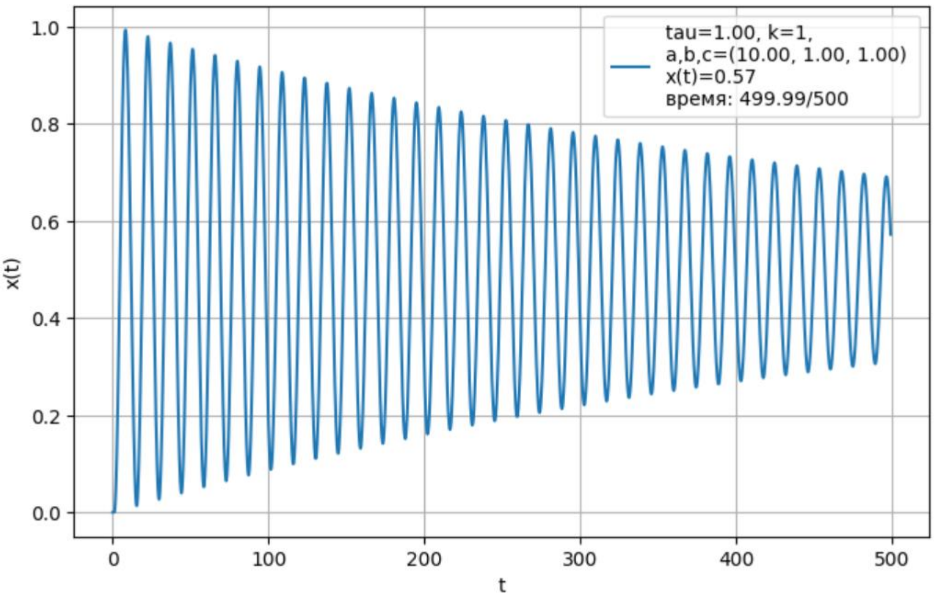

Рисунок 3 - График состояния системы при задержке в 100 шагов и а=10. Несмотря на большую амплитуду колебаний, система рано или поздно должна сойтись.

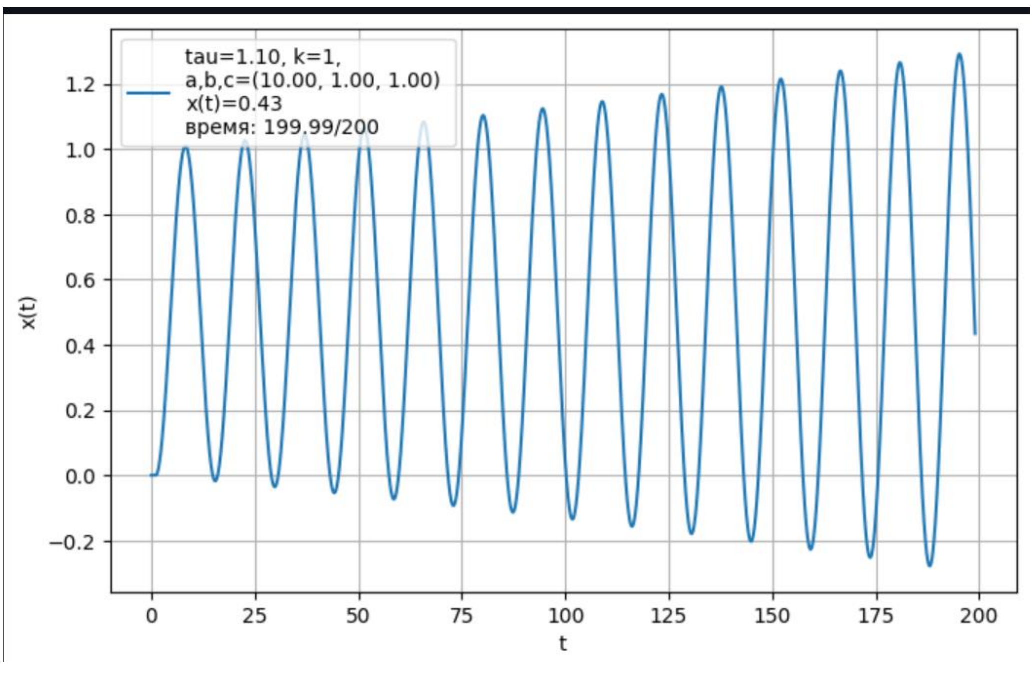

Рисунок 4 - График состояния системы при увеличении задержки.

В отличие от прошлого рисунка, задержка привела к расхождению. Из этого можно сделать вывод, что граничное значение запаздывания в текущей конфигурации равно примерно 100 шагов. Однако при уменьшении параметра *a* до 2, система начинает сходиться даже при задержке в 110 шагов, что говорит об обратной зависимости величины *a* по отношению к значению граничного запаздывания. Подобное поведение можно наблюдать на рисунке 5

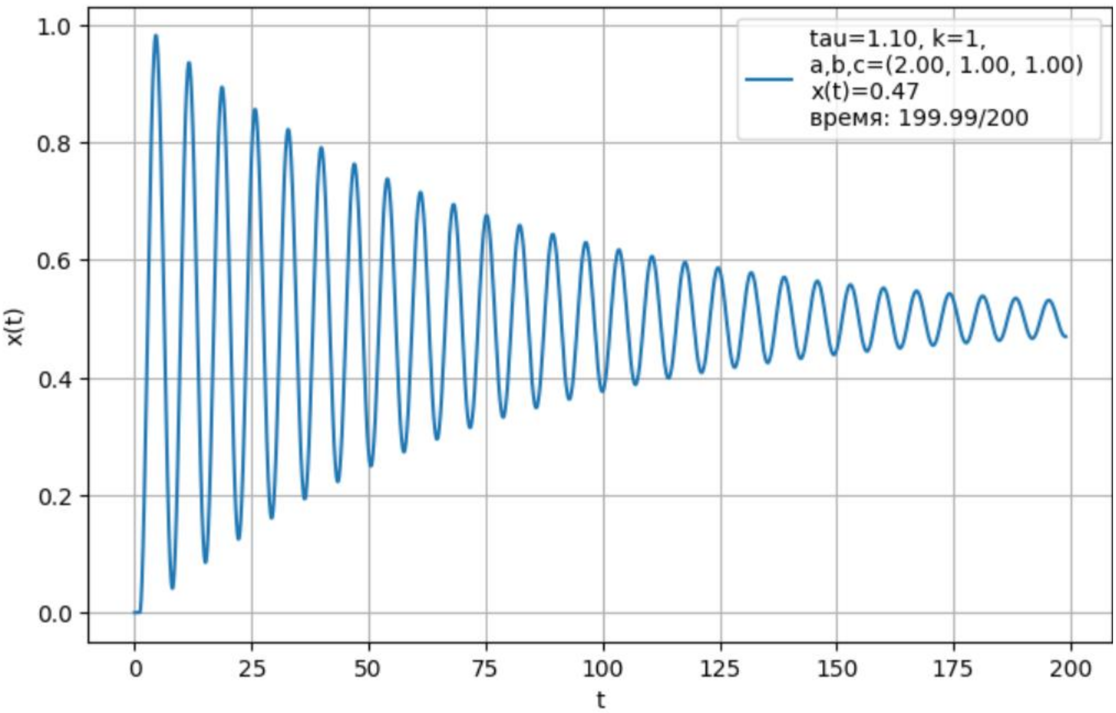

Рисунок 5 - Сходимость среды при уменьшении параметра *a*

Вернемся к конфигурации tau=110, *a*=10, и попробуем уменьшить параметр *k*. Как можно увидеть на рисунке 6, это привело к сходимости среды, так что можно сделать вывод, что параметр *k* должен быть меньше 1 в случае

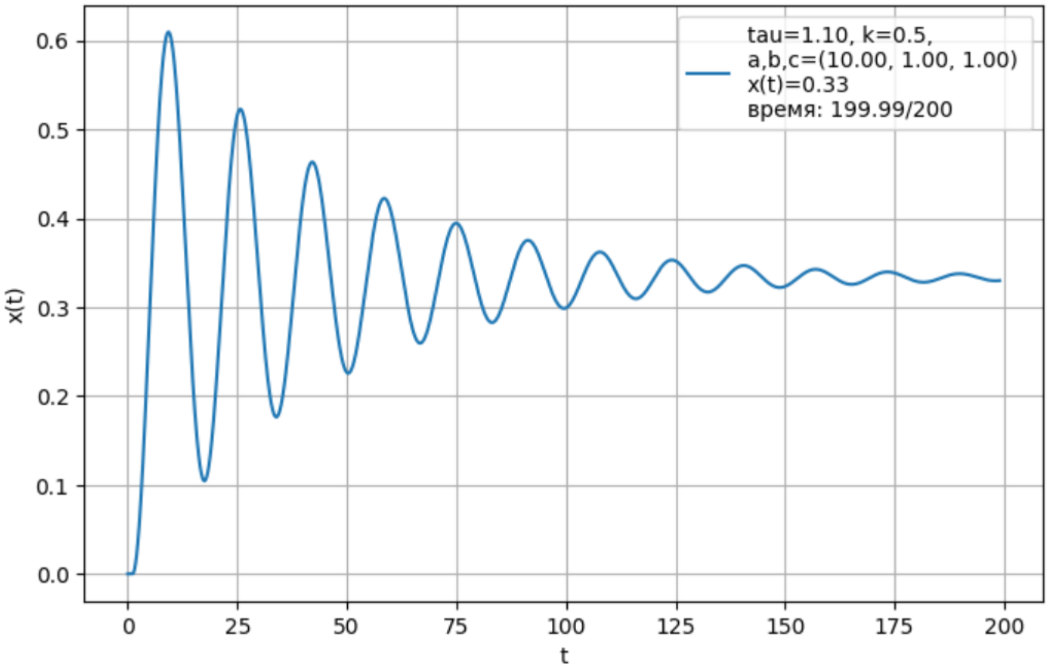

Рисунок 6 - Уменьшение параметра *k* привело к сходимости даже при большой задержке и большом значении параметра *a*.

Теперь рассмотрим влияние параметра *c* на систему. Можно отметить, что любое изменение, приводит к сходимости, даже при довольно большой задержке. Увеличение параметра *c* приводит систему в устойчивое состояние при большой задержке, но в малой окрестности нуля, а уменьшение параметра *c* приводит к сходимости в 1, уменьшая граничное значение запаздывания. Соответствующие графики изображены на рисунках 7 и 8.

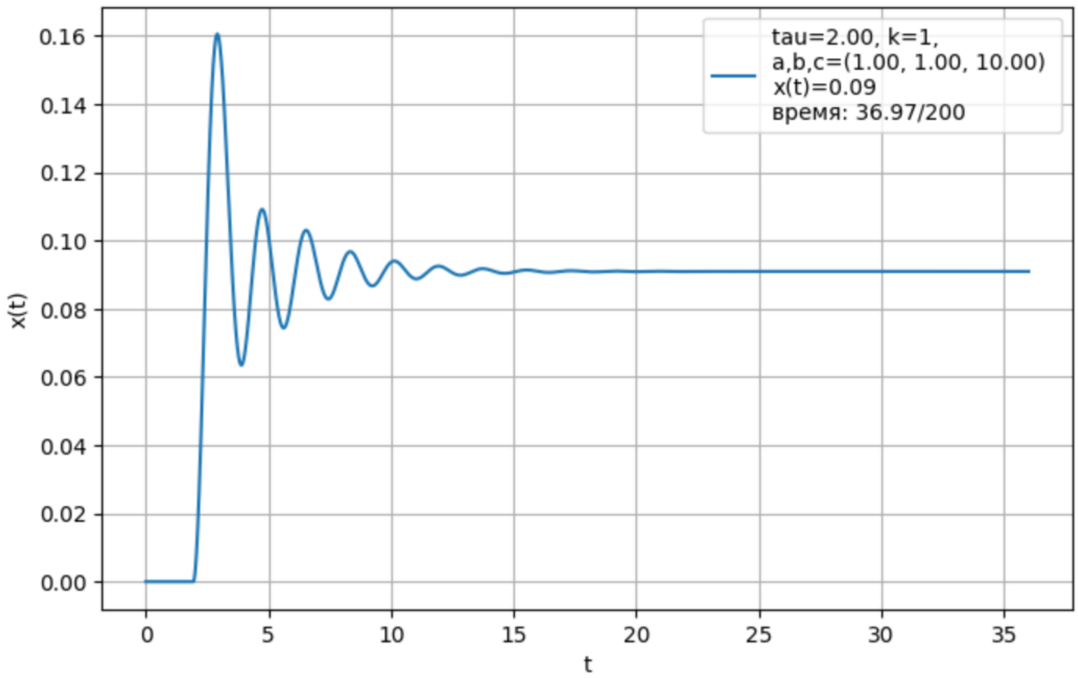

Рисунок 7 - Увеличение параметра *c* при большой задержке приводит к сходимости, но с отклонением.

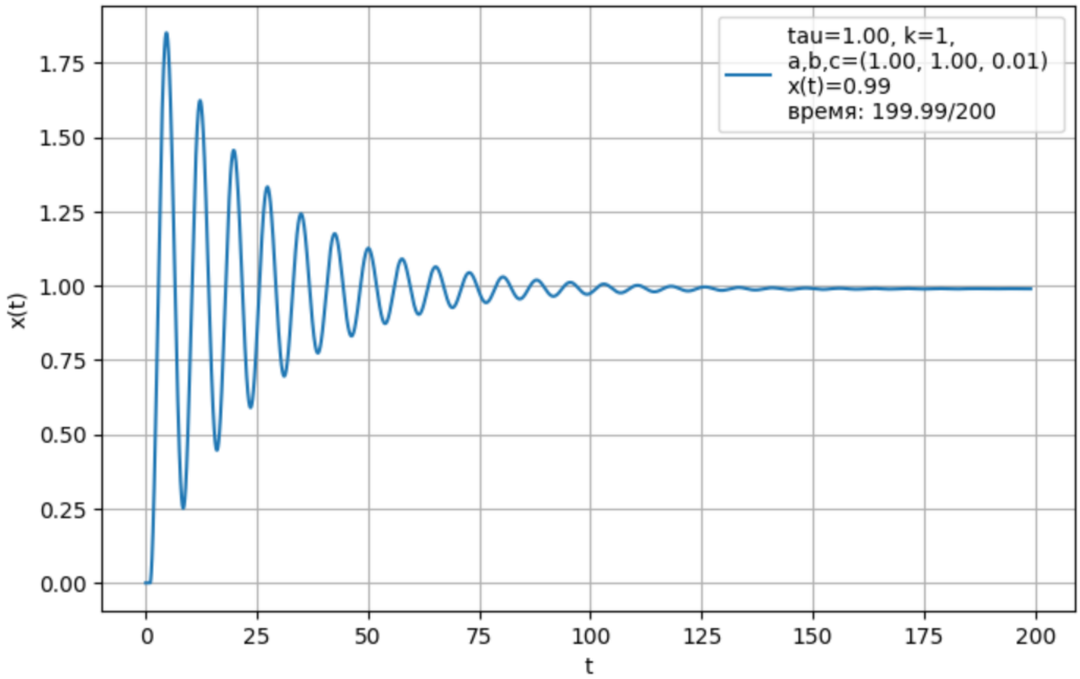

Рисунок 8 - Уменьшение параметра *c* уточняет целевую функцию, но уменьшает порог устойчивости(задержка примерно 100-110)

Выводы: по результатам запусков можно сделать вывод, что в динамических системах важен детальный подбор параметров. Так, в нашем случае можно предположить:

- увеличение *a* уменьшает граничное значение запоздания и увеличивает время схождения
- увеличивание b увеличивает граничное значение запоздания и уменьшает время схождения
- увеличивание c увеличивает граничное значение запоздания
- увеличивание k уменьшает граничное значение запоздания