

Санкт-Петербургский государственный университет

Прикладная математика, информатика и искусственный интеллект

Отчет по учебной практике 4 (семестр 4)

Научная исследовательская работа

ПОГРАНИЧНЫЙ СЛУЧАЙ УСТОЙЧИВОСТИ СИСТЕМЫ ФАПЧ

Выполнил:

Кривоносов Тимофей Игоревич, группа
22Б05-мм



Научный руководитель:

М. Ю. Лобачёв

Ассистент



Кафедра ПРИКЛАДНОЙ

КИБЕРНЕТИКИ Санкт-Петербургий

государственный университет

Математико-механический факультет

Работа выполнена полностью и в срок.
Оценка А

Санкт-Петербург

2024

Схемы фазовой автоподстройки частоты (ФАП) являются нелинейными системами управления, которые были изобретены в 1930-х годах и получили широкое распространение в системах связи, распределенных компьютерных архитектурах, глобальных навигационных спутниковых системах, энергетических системах и других приложениях. В данной работе проводится анализ предельных режимов дискретной ФАП второго порядка. Работа выполнена полностью и в срок, все поставленные задачи выполнены. Необходимые компетенции согласно программе дисциплины и образовательной программы сформированы. Оценка отлично (А).

ассистент математико-механического факультета

 Лобачев М.Ю.

Оглавление

1.	ОТЗЫВ	2
2.	ВВЕДЕНИЕ	4
3.	АНАЛИЗ УСТОЙЧИВОСТИ	5
4.	МОДЕЛИРОВАНИЕ	10
4.1.	Краткое пояснение кода программы	10
4.2.	Результаты моделирования	14
5.	ЗАКЛЮЧЕНИЕ	15
6.	ИСТОЧНИКИ	16
7.	ПРИЛОЖЕНИЯ	16

2. ВВЕДЕНИЕ

В данной научно-исследовательской работе изучается пограничный случай устойчивости системы фазовой автоподстройки частоты.

Целью данной работы является нахождение условий, при которых система сохраняет устойчивое поведение при различных параметрах.

Актуальность исследования обусловлена широким применением систем фазовой автоподстройки частоты в современных телекоммуникационных и радиотехнических устройствах. Системы ФАПЧ используются для синхронизации частот в различных приложениях, начиная от радиоприемников и заканчивая сложными системами передачи данных, такими как спутниковая связь и мобильные сети. Надежная работа таких систем зависит от их устойчивости при различных

Таким образом, данная работа вносит вклад в теоретическое изучение и практическое применение теории устойчивости для сложных динамических систем, таких как системы фазовой автоподстройки частоты.

3. АНАЛИЗ УСТОЙЧИВОСТИ

Подробнее ознакомиться с системой ФАПЧ, можно в работах:

- *"The Egan Problem on the Pull-in Range of Type 2 PLLs"* Nikolay V. Kuznetsov , Mikhail Y. Lobachev, Marat V. Yuldashev, and Renat V. Yuldashev. 6
- *"Nonlinear Systems Analysis."* Egan, W. F. (1969), Englewood Cliffs, NJ: Prentice-Hall.

В данной работе акцент поставлен на изучение устойчивости.

$$\begin{aligned}\dot{x}_1 &= \sin \theta_e \\ \dot{x}_2 &= -\frac{1}{\tau_p} x_2 + \frac{(\tau_{z1} - \tau_p)(\tau_p - \tau_{z2})}{\tau_p^2} \sin \theta_e \\ \dot{\theta}_e &= \omega_e^{free} - K(x_1 + x_2 + \frac{\tau_{z1}\tau_{z2}}{\tau_p} \sin \theta_e)\end{aligned}$$

- найти состояние равновесия.
- Провести линеаризацию в окрестности и выписать характеристический полином.
- При $\chi_J(i\omega) \neq 0$ определить с помощью алгебраических критериев условия асимптотической устойчивости с.р (в зависимости от параметров:

$$K > 0, \tau_{z1} > 0, \tau_{z2} > 0, \tau_p > 0, \tau_{z1} \neq \tau_p, \tau_{z2} \neq \tau_p \quad (1)$$

- определить устойчивость состояний равновесия в случае $\chi_J(i\omega)$ с помощью теоремы Ляпунова.

1 Найти с.р

Найдем вектор, который при подставлении в систему обращает ее в нуль. Для этого приравняем правые части к нулю.

Рассмотрим

$$\sin \theta_e = 0$$

Из этого следует, что $\theta = \pi k$, где $k \in \mathbb{Z}$

Подставим $\theta = \pi k$ в систему:

$$-\frac{1}{\tau_p}x_2 = 0$$

$$\omega_e^{free} - K(x_1 + x_2) = 0$$

Из первого уравнения получим $x_2 = 0, \tau_p \neq 0$

Подставив во второе уравнение, получим: $\omega_e^{free} - Kx_1$, откуда $x_1 = \frac{\omega_e^{free}}{K}$

Получили вектор:

$$\begin{pmatrix} \frac{\omega_e^{free}}{K} \\ 0 \\ \pi k \end{pmatrix}$$

2 Провести линеаризацию в окрестности и выписать х.п

Выпишем матрицу Якоби исходной системы:

$$\begin{pmatrix} 0 & 0 & \cos(\theta_e) \\ 0 & -\frac{1}{\tau_p} & \frac{(\tau_{z1}-\tau_p)(\tau_p-\tau_{z2})}{\tau_p^2} \cos(\theta_e) \\ -K & -K & -K \frac{\tau_{z1}\tau_{z2}}{\tau_p} \cos(\theta_e) \end{pmatrix}$$

Заметим, что матрица Якоби не зависит от первых двух компонент вектора состояний равновесия, а $\cos(\theta_e)$ на решениях вида πk принимает ровно два значения: 1 и -1 , потому рассмотрим два этих случая и найдем характеристические полиномы:

1. При $\theta_e = 2\pi k$ имеем:

$$\chi(\lambda) = \begin{vmatrix} -\lambda & 0 & 1 \\ 0 & -\frac{1}{\tau_p} - \lambda & \frac{(\tau_{z1}-\tau_p)(\tau_p-\tau_{z2})}{\tau_p^2} \\ -K & -K & -K \frac{\tau_{z1}\tau_{z2}}{\tau_p} - \lambda \end{vmatrix}$$

А в общем виде:

$$\chi(\lambda) = (-1)(\lambda^3 + \lambda^2(\frac{1 + K\tau_{z1}\tau_{z2}}{\tau_p}) + \lambda K(\frac{\tau_{z1} + \tau_{z2}}{\tau_p}) + \frac{K}{\tau_p})$$

2. При $\theta_e = \pi + 2\pi k$ имеем:

$$\chi(\lambda) = \begin{vmatrix} -\lambda & 0 & -1 \\ 0 & -\frac{1}{\tau_p} - \lambda & -\frac{(\tau_{z1}-\tau_p)(\tau_p-\tau_{z2})}{\tau_p^2} \\ -K & -K & K \frac{\tau_{z1}\tau_{z2}}{\tau_p} - \lambda \end{vmatrix}$$

А в общем виде:

$$\chi(\lambda) = (-1)(\lambda^3 + \lambda^2(\frac{1 - K\tau_{z_1}\tau_{z_2}}{\tau_p}) - \lambda K(\frac{\tau_{z_1} + \tau_{z_2}}{\tau_p}) + \frac{K}{\tau_p})$$

3 Критерии асимптотической устойчивости с.р $\chi_J(i\omega) \neq 0$

Параметры заданные условием: $K > 0, \tau_{z_1} > 0, \tau_{z_2} > 0, \tau_p > 0;$

$\tau_{z_1} \neq \tau_p > 0, \tau_{z_2} \neq \tau_p > 0.$

1. При $\theta_e = \pi + 2\pi k$ имеем:

$$\chi(\lambda) = (-1)(\lambda^3 + \lambda^2(\frac{1 - K\tau_{z_1}\tau_{z_2}}{\tau_p}) - \lambda K(\frac{\tau_{z_1} + \tau_{z_2}}{\tau_p}) + \frac{K}{\tau_p})$$

Заметим, что $-K\frac{\tau_{z_1} + \tau_{z_2}}{\tau_p}, \frac{K}{\tau_p}$ - разных знаков. Это противоречит необходимому условию устойчивости по Раусу. Отсюда следует, что в условии задачи с.р неустойчиво при любых допустимых значениях.

2. При $\theta_e = 2\pi k$ имеем:

$$\chi(\lambda) = (-1)(\lambda^3 + \lambda^2(\frac{1 + K\tau_{z_1}\tau_{z_2}}{\tau_p}) + \lambda K(\frac{\tau_{z_1} + \tau_{z_2}}{\tau_p}) + \frac{K}{\tau_p})$$

Используем следствие из критерия Рауса-Гурвица:

Следствие: Полином $\chi(s) = s^3 + a_1s^2 + a_2s + a_3$ устойчив $\Leftrightarrow a_{1,2,3} > 0, a_1a_2 > a_3.$

- $\frac{1 + K\tau_{z_1}\tau_{z_2}}{\tau_p}, K\frac{\tau_{z_1} + \tau_{z_2}}{\tau_p}, \frac{K}{\tau_p} > 0$ в силу условий задачи.
- Проверим второе неравенство:

$$\frac{1 + K\tau_{z_1}\tau_{z_2}}{\tau_p} * K\frac{\tau_{z_1} + \tau_{z_2}}{\tau_p} > \frac{K}{\tau_p}$$

Т.к $\tau_p > 0$ имеем такое условие:

$$\frac{\tau_p - (\tau_{z_1} + \tau_{z_2})}{(\tau_{z_1} + \tau_{z_2})\tau_{z_1}\tau_{z_2}} < K$$

4 Случай $\chi(i\omega) = 0$

Многочлен 3-й степени с действительными коэффициентами $q(\lambda) = \lambda^3 + a\lambda^2 + b\lambda + c$ при $a, b, c > 0$ имеет чисто мнимый корень тогда и только тогда, когда $ab = c$, следовательно, чтобы

$\chi(\lambda) = (-1)(\lambda^3 + \lambda^2(\frac{1+K\tau_{z1}\tau_{z2}}{\tau_p}) + \lambda K(\frac{\tau_{z1}+\tau_{z2}}{\tau_p}) + \frac{K}{\tau_p})$ имел чисто мнимый корень необходимо и достаточно, чтобы:

$$\begin{cases} \tau_p - (\tau_{z1} + \tau_{z2}) > 0 \\ \frac{\tau_p - (\tau_{z1} + \tau_{z2})}{(\tau_{z1} + \tau_{z2})\tau_{z1}\tau_{z2}} = K \end{cases}$$

Найдем собственные числа $\chi(\lambda) = (-1)(\lambda^3 + \lambda^2(\frac{1+K\tau_{z1}\tau_{z2}}{\tau_p}) + \lambda K(\frac{\tau_{z1}+\tau_{z2}}{\tau_p}) + \frac{K}{\tau_p})$ для этого воспользуемся формулой Кардано для $ax^3 + bx^2 + cx + d = 0, a \neq 0$.

Рассмотрим замены:

$$\begin{aligned} \lambda &= y - \frac{b}{3a} \\ p &= \frac{3ac - b^2}{3a^2} \quad q = \frac{2b^3 - 9abc + 27a^2d}{27a^3} \\ t_{1,2} &= \sqrt[3]{-\frac{q}{2} \pm \sqrt{\frac{q^2}{2} + \frac{p^3}{3}}} \\ \lambda &= t_1 + t_2 - \frac{b}{3a} \end{aligned}$$

$$a = 1 \quad b = \frac{1 + K\tau_{z1}\tau_{z2}}{\tau_p} \quad c = \frac{\tau_{z1} + \tau_{z2}}{\tau_p} \quad d = \frac{K}{\tau_p}$$

Тогда подставляя в полученные выше выражения, данные характеристического полинома. Получим равенство на λ , для разных значений корня.

$$\begin{aligned} \lambda &= t_1 + t_2 - \frac{1 + K\tau_{z1}\tau_{z2}}{3\tau_p} \\ \lambda_{1,2,3} &= \sqrt[3]{-\frac{q}{2} + \sqrt{\frac{q^2}{2} + \frac{p^3}{3}}} + \sqrt[3]{-\frac{q}{2} - \sqrt{\frac{q^2}{2} + \frac{p^3}{3}}} - \frac{1 + K\tau_{z1}\tau_{z2}}{3\tau_p} \\ p &= \frac{3(\frac{\tau_{z1}+\tau_{z2}}{\tau_p}) - (\frac{1+K\tau_{z1}\tau_{z2}}{\tau_p})^2}{9} \\ q &= \frac{2(\frac{1+K\tau_{z1}\tau_{z2}}{\tau_p})^3 - 9(\frac{1+K\tau_{z1}\tau_{z2}}{\tau_p})(\frac{\tau_{z1}+\tau_{z2}}{\tau_p}) + 27\frac{K}{\tau_p}}{27} \end{aligned}$$

Для анализа случая, когда характеристический полином системы $\chi(i\omega) = 0$ была смоделирована. Моделирование проводилось с использованием Jupyter Notebook, в котором были выполнены численные расчеты и построены графики, иллюстрирующие динамическое поведение системы. Основываясь на полученных Рис. 3.

результатах, можно сделать вывод, что в представленном граничном случае состояние равновесия неустойчиво.

4. МОДЕЛИРОВАНИЕ

В данном разделе рассмотрена модель системы с целью анализа её устойчивости. Использование численных методов и визуализация фазовых портретов позволяет лучше понять динамическое поведение системы.

4.1. Краткое пояснение кода программы

Для моделирования необходимы такие библиотеки:

```
import numpy as np
import matplotlib.pyplot as plt
import math
from mpl_toolkits.mplot3d import Axes3D
from scipy.integrate import odeint
```

Для проверки этих условий полученных в пункте 3 используется следующая функция:

```
def checker(tau_p, tau_z1, tau_z2, omega_free):
    if (tau_p - (tau_z1 + tau_z2) <= 0):
        print("Condition_tau_p_-(tau_z1+_tau_z2)_>_0_not_met")
        return False

    if (tau_p == tau_z2):
        print("Condition_tau_p!=_tau_z2_not_met")
        return False

    if (tau_p == tau_z1):
        print("Condition_tau_p!=_tau_z1_not_met")
        return False

    if (tau_z1 <= 0 or tau_z2 <= 0 or tau_p <= 0 or omega_free <= 0):
        print("Parameters_<_0")
        return False

    return True
```

Заданные параметры системы и вычисление K :

Параметры системы:

$$\tau_{z1} = 0.5$$

$$\tau_{z2} = 0.3$$

$$\tau_p = 0.9$$

$$\omega_{free} = 0.99$$

Вычисление K по формуле

```
K = (tau_p - (tau_z1 + tau_z2)) / ((tau_z1 + tau_z2) * tau_z1 * tau_z2)
print (K)
cheker(tau_p, tau_z1, tau_z2, omega_free)
```

Зададим интервалы для моделирования:

```
grapg_x = [-0.1, 0.1]
grapg_y = [-0.1, 0.1]
grapg_teta = [0, math.pi]
```

Система дифференциальных уравнений задается функцией:

```
def system(state, t):
    x1, x2, theta = state
    dx1_dt = np.sin(theta)
    dx2_dt = -(1/tau_p) * x2 + ((tau_z1 - tau_p) * (tau_p - tau_z2) /
        tau_p**2) * np.sin(theta)
    dtheta_dt = omega_free - K * (x1 + x2 + (tau_z1 * tau_z2 / tau_p) *
        np.sin(theta))
    return [dx1_dt, dx2_dt, dtheta_dt]
```

Для построения фазового портрета используется следующая функция:

```
def plot_phase_portrait():
    t = np.linspace(0, 50, 1000) # Increase the time interval and the
        number of points
    # Generate a set of initial conditions
    num_initial_conditions = 7
    x1_initials = np.linspace(grapg_x[0], grapg_x[1],
        num_initial_conditions)
```

```

x2_initials = np.linspace(grapg_y[0], grapg_y[1],
    num_initial_conditions)
theta_initials = np.linspace(grapg_teta[0], grapg_teta[1],
    num_initial_conditions)

fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')

for x1 in x1_initials:
    for x2 in x2_initials:
        for theta in theta_initials:
            initial = [x1, x2, theta]
            solution = odeint(system, initial, t)
            x1_sol, x2_sol, theta_sol = solution.T
            ax.plot(x1_sol, x2_sol, theta_sol, alpha=0.7)

ax.set_xlabel('x1')
ax.set_ylabel('x2')
ax.set_zlabel('theta')
ax.set_title('Phase_Portrait_of_the_System')
ax.view_init(elev=20, azim=30)
plt.show()

```

```

# Call the function to plot the phase portrait
plot_phase_portrait()

```

Функция для вычисления собственных чисел:

```

def compute_eigenvalues(K, tau_z1, tau_z2, tau_p):
    coeffs = [1, (1 + K * tau_z1 * tau_z2) / tau_p, K * (tau_z1 +
        tau_z2) / tau_p, K / tau_p]
    eigenvalues = np.roots(coeffs)
    return eigenvalues

eigenvalues = compute_eigenvalues(K, tau_z1, tau_z2, tau_p)
print(f'Roots:{eigenvalues[0],eigenvalues[1],eigenvalues[2]}')

```

В библиотеке `scipy.integrate`, функция `odeint` используется для численного решения систем обыкновенных дифференциальных уравнений (ODEs). Она основана на методах с переменным шагом и порядком, разработанных для решения жестких и нежестких систем ODE.

Порядок ошибки на шаге: // Для нежестких систем ODE `odeint` использует метод интеграции LSODA (Livermore Solver for Ordinary Differential equations with Automatic method switching for stiff and non-stiff problems). Этот метод комбинирует два алгоритма:

- **Adams-Bashforth-Moulton метод (non-stiff):** Это метод переменного порядка (до 12-го порядка). Для нелинейных уравнений это метод порядка $O(h^p)$, где h — шаг интегрирования, а p — порядок метода, который может изменяться в зависимости от условий интеграции.
- **Backward Differentiation Formula (BDF, stiff):** Это также метод переменного порядка (до 5-го порядка), и его ошибка на шаге составляет $O(h^p)$, где p — порядок метода, который также может изменяться.

Суммарная ошибка

Суммарная ошибка интеграции определяется не только ошибкой на одном шаге, но и накоплением ошибок на протяжении всех шагов интеграции. В идеальном случае, суммарная ошибка примерно пропорциональна $O(h^{p-1})$, где p — порядок метода.

Фазовый портрет системы позволяет наглядно оценить её динамическое поведение и устойчивость при различных начальных условиях и параметрах.

4.2. Результаты моделирования

Запустив модель получили такие результаты:

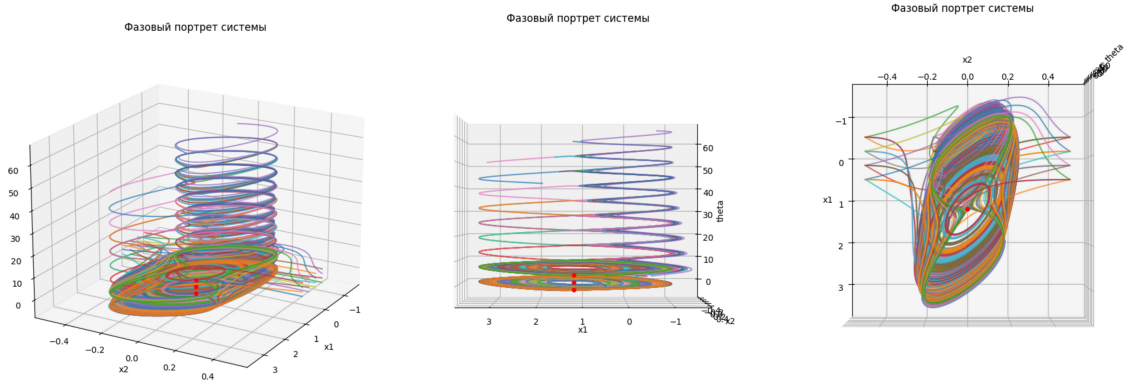


Рис. 1. Модель: $\tau_{z1} = 0.5$, $\tau_{z2} = 0.3$, $\tau_p = 0.9$, $\omega_{free} = 0.99$, $K = 0.8333331$, $\text{graph_x} = [-0.1, 0.1]$, $\text{graph_y} = [-0.1, 0.1]$, $\text{graph_theta} = [0, \pi]$

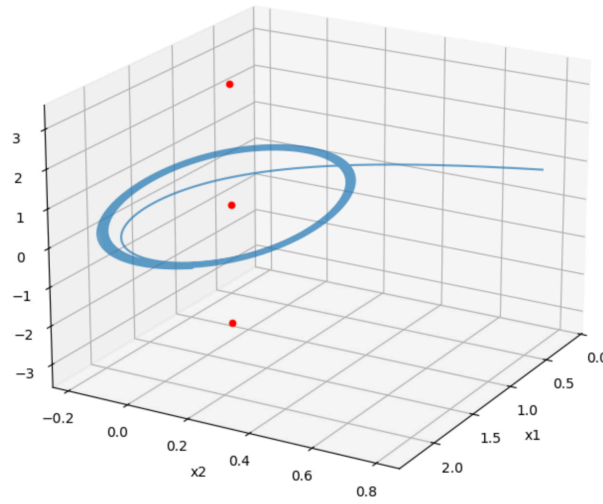


Рис. 2. Одна траектория

Заметим что в силу: $\frac{\tau_p - (\tau_{z1} + \tau_{z2})}{(\tau_{z1} + \tau_{z2})\tau_{z1}\tau_{z2}} = K$, K обратно пропорционально $\tau_{z1}\tau_{z2}(\tau_{z1} - \tau_{z2})$, поэтому уменьшая порядок $\tau_{z1}\tau_{z2}$ на 10^{-1} значение K увеличивается в 1000 раз. Отсюда такая большая разница между графиками Рис. 2. и Рис. 3.

Основываясь на полученных результатах, можно сделать вывод, что в представленных граничных случаях состояния равновесия неустойчивы.

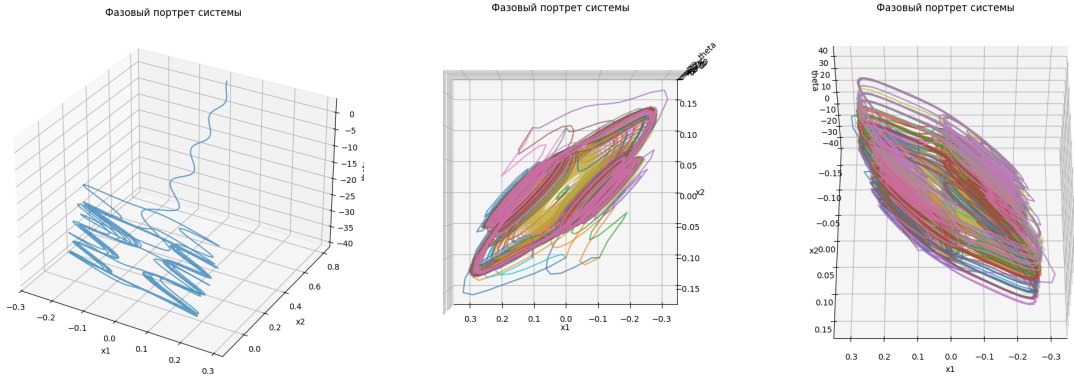


Рис. 3. Модель: $\tau_{z1} = 0.1$, $\tau_{z2} = 0.1$, $\tau_p = 0.4$, $\omega_{free} = 1$, $K = 99.9999999$, $\text{graph_x} = [-0.1, 0.1]$, $\text{graph_y} = [-0.1, 0.1]$, $\text{graph_theta} = [0, \pi]$

5. ЗАКЛЮЧЕНИЕ

В ходе работы достигнуты поставленные цели, исследована асимптотическая устойчивость решений системы ФАПЧ. Использованы фундаментальные теоремы и принципы теории устойчивости, включая линеаризацию систем и критерий Рауса-Гурвица, чтобы определить условия устойчивости.

Для моделирования использованы численные методы, реализованные в среде программирования Python. Написаны программы для решения ОДУ и построения фазовых портретов системы. В результате моделирования получены данные, позволившие проанализировать поведение системы при различных параметрах.

Проведя анализ устойчивости, определены состояние равновесия системы:

$$\begin{pmatrix} \frac{w_e^{free}}{K} \\ 0 \\ \pi k \end{pmatrix},$$

где $k \in \mathbb{Z}$. Для параметров системы $K > 0$, $\tau_{z1} > 0$, $\tau_{z2} > 0$, $\tau_p > 0$, $\tau_{z1} \neq \tau_p$, $\tau_{z2} \neq \tau_p$ были установлены условия устойчивости согласно критерию Рауса-Гурвица. Полином $\chi(\lambda) = \lambda^3 + \lambda^2 \left(1 + \frac{K\tau_{z1}\tau_{z2}}{\tau_p}\right) + \lambda K \left(\frac{\tau_{z1} + \tau_{z2}}{\tau_p}\right) + \frac{K}{\tau_p}$ устойчив при выполнении неравенств: $1 + \frac{K\tau_{z1}\tau_{z2}}{\tau_p} > 0$, $\frac{K(\tau_{z1} + \tau_{z2})}{\tau_p} > 0$, и $\frac{K}{\tau_p} > 0$. Проверка второго неравенства дает условие: $K < \frac{\tau_p - (\tau_{z1} + \tau_{z2})}{(\tau_{z1} + \tau_{z2})\tau_{z1}\tau_{z2}}$.

Таким образом, результаты моделирования и анализа устойчивости показали, что состояние равновесия системы может быть как устойчивым, так и неустойчивым в зависимости от параметров системы.

6. ИСТОЧНИКИ

- *"The Egan Problem on the Pull-in Range of Type 2 PLLs"* Nikolay V. Kuznetsov , Mikhail Y. Lobachev, Marat V. Yuldashev, and Renat V. Yuldashev: [Online]. Available: <https://ieeexplore.ieee.org/document/9258948>
- *"ХАОТИЧЕСКАЯ ДИНАМИКА И КЛАССИЧЕСКАЯ ТЕОРИЯ УСТОЙЧИВОСТИ ДВИЖЕНИЯ"* Г. А. Леонов [Online]. Available: <https://math.spbu.ru/user/leonov/publications/chaos/chaos.pdf>
- *"Nonlinear Systems Analysis."* Egan, W. F. (1969), Englewood Cliffs, NJ: Prentice-Hall. [Online]. Available: <https://wp.kntu.ac.ir/nobari/Nonlinear/VIDIASAGAR.pdf>
- *"Открытый искусственный интеллект"* [Online]. Available: <https://openai.com>

7. ПРИЛОЖЕНИЯ

```
import numpy as np
import matplotlib.pyplot as plt
import math
from mpl_toolkits.mplot3d import Axes3D
from scipy.integrate import odeint

def checker(tau_p, tau_z1, tau_z2, omega_free):
    if (tau_p - (tau_z1 + tau_z2) <= 0):
        print("Condition_tau_p_-(tau_z1+_tau_z2)_>_0_not_met")
        return False

    if (tau_p == tau_z2):
        print("Condition_tau_p!=_tau_z2_not_met")
        return False

    if (tau_p == tau_z1):
        print("Condition_tau_p!=_tau_z1_not_met")
```



```

        return False

    if (tau_z1 <= 0 or tau_z2 <= 0 or tau_p <= 0 or omega_free <= 0):
        print("Parameters_<_0")
        return False

    return True

# System parameters
tau_z1 = 0.5
tau_z2 = 0.3
tau_p = 0.9
omega_free = 0.99

# Calculate K using the formula
K = (tau_p - (tau_z1 + tau_z2)) / ((tau_z1 + tau_z2) * tau_z1 * tau_z2)
print(K)
checker(tau_p, tau_z1, tau_z2, omega_free)

graph_x = [-0.1, 0.1]
graph_y = [-0.1, 0.1]
graph_theta = [0, math.pi]

graph_y[1]

# System of differential equations
def system(state, t):
    x1, x2, theta = state
    dx1_dt = np.sin(theta)
    dx2_dt = -(1/tau_p) * x2 + ((tau_z1 - tau_p) * (tau_p - tau_z2) /
        tau_p**2) * np.sin(theta)
    dtheta_dt = omega_free - K * (x1 + x2 + (tau_z1 * tau_z2 / tau_p) *
        np.sin(theta))
    return [dx1_dt, dx2_dt, dtheta_dt]

```

```

# Function to plot the phase portrait
def plot_phase_portrait():
    t = np.linspace(0, 50, 1000) # Increase the time interval and the
        number of points

    # Generate a set of initial conditions
    num_initial_conditions = 7
    x1_initials = np.linspace(graph_x[0], graph_x[1],
        num_initial_conditions)
    x2_initials = np.linspace(graph_y[0], graph_y[1],
        num_initial_conditions)
    theta_initials = np.linspace(graph_theta[0], graph_theta[1],
        num_initial_conditions)

    fig = plt.figure(figsize=(12, 8))
    ax = fig.add_subplot(111, projection='3d')

    for x1 in x1_initials:
        for x2 in x2_initials:
            for theta in theta_initials:
                initial = [x1, x2, theta]
                solution = odeint(system, initial, t)
                x1_sol, x2_sol, theta_sol = solution.T
                ax.plot(x1_sol, x2_sol, theta_sol, alpha=0.7)

    ax.set_xlabel('x1')
    ax.set_ylabel('x2')
    ax.set_zlabel('theta')
    ax.set_title('Phase_Portrait_of_the_System')
    ax.view_init(elev=20, azim=30)
    plt.show()

# Call the function to plot the phase portrait

```

```

plot_phase_portrait()

# Initial conditions and time interval
initial_conditions = [0.3, 0.8, np.pi/3] # x1, x2, theta
t = np.linspace(0, 50, 10000) # time interval

def plot_phase_portrait(initial_conditions, t):
    fig = plt.figure(figsize=(12, 8))
    ax = fig.add_subplot(111, projection='3d')

    solution = odeint(system, initial_conditions, t)
    x1_sol, x2_sol, theta_sol = solution.T
    ax.plot(x1_sol, x2_sol, theta_sol, alpha=0.7)

    ax.set_xlabel('x1')
    ax.set_ylabel('x2')
    ax.set_zlabel('theta')
    ax.set_title('Phase_Portrait_of_the_System')
    # ax.view_init(elev=20, azimuth=30)
    plt.show()

# Call the function to plot the phase portrait
plot_phase_portrait(initial_conditions, t)

```