Национальный исследовательский университет «МЭИ»

ЛАБОРАТОРНАЯ РАБОТА №3

«ОПРЕДЕЛЕНИЕ НАСТРОЕЧНЫХ ПАРАМЕТРОВ РЕГУЛЯТОРА МЕТОДОМ РУЧНОГО ПОДБОРА»

Выполнил: Максимов Р.С.

Москва 2019 г.

Оглавление

[2. Задание 3](#_Toc27955327)

[3. Исходные данные 3](#_Toc27955328)

[4. Анализ параметров ПИД-регулятора 3](#_Toc27955329)

[a. Прямой метод 3](#_Toc27955330)

[b. Корневой метод 4](#_Toc27955331)

[с. По логарифмическим частотным характеристикам 5](#_Toc27955332)

[d. Интегральный метод 6](#_Toc27955333)

[5. Анализ параметров ПИ-регулятора 7](#_Toc27955334)

[a. Прямой метод 7](#_Toc27955335)

[b. Корневой метод 8](#_Toc27955336)

[с. По логарифмическим частотным характеристикам 9](#_Toc27955337)

[e. Интегральный метод 9](#_Toc27955338)

[Вывод: 10](#_Toc27955339)

[Приложение А: код программы. 11](#_Toc27955340)

#### **Цель работы**

Определение показателей качества стационарной системы автоматического управления в установившихся и переходных режимах прямыми и косвенными методами.

1. **Задание**

1. Необходимо определить передаточную функцию АСР

2. Определить параметры регулятора для соответствия переходного процесса заданным критериям качества, приведенных в таблице 1.

3. Определить прямые оценки качества переходного процесса:

4. По распределению корней на комплексной плоскости замкнутой САУ определить.

5. По логарифмическим частотным характеристикам определить.

6. По интегральному методу оценить степень качества системы регулирования

1. **Исходные данные**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Вариант | Время регулирования, с | Показатель колебательности | Перерегулирование,% | Тип регулятора |
| 14 | 13 | 1.20 | 25 | ПИД, ПИ |

1. **Анализ параметров ПИД-регулятора**
   1. **Прямой метод**

На рисунке 1 представлена структурная схема САУ. Для данной схемы на рисунке 2 представлена переходная характеристика.



Рисунок 1. Структурная схема АСР

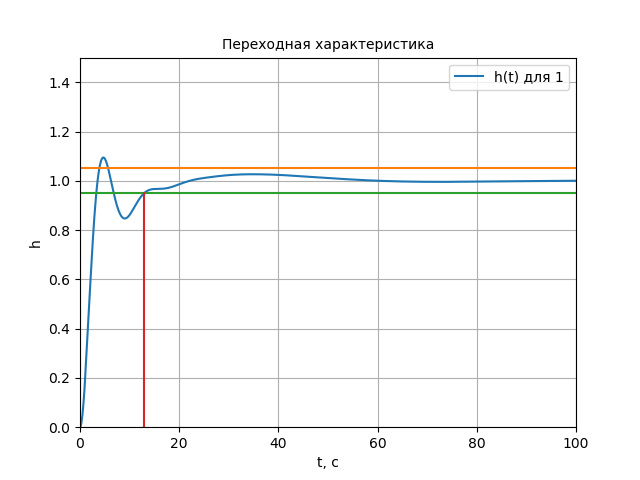
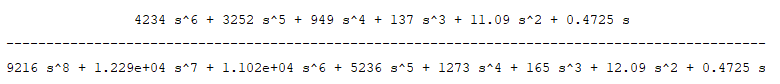
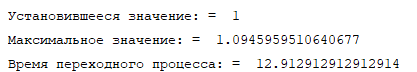


Рисунок 2. Переходная характеристика.

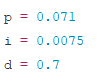
Передаточная функция:



Вывод программы:



Параметры ПИД-регулятора:



* 1. **Корневой метод**

На рисунке 3 представлено распределение корней характеристического уравнения на координатной плоскости:

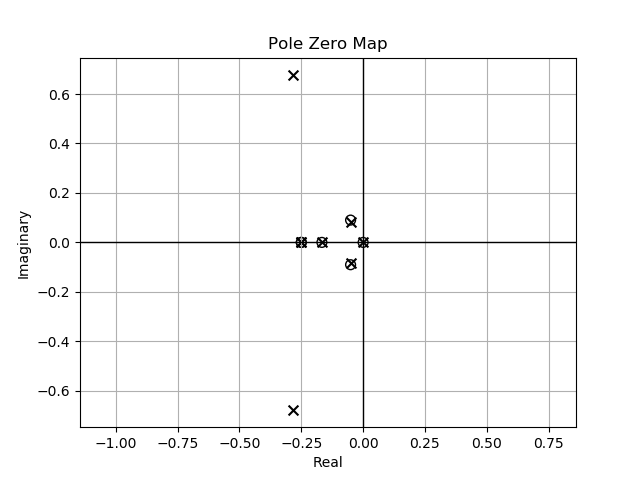
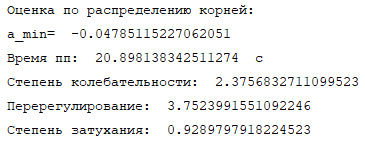


Рисунок 3- Корни характеристического уравнения.

Вывод программы:



**с. По логарифмическим частотным характеристикам**

На рисунке 4 представлены частотные характеристики для данной САУ (ЛАЧХ и ЛФЧХ).

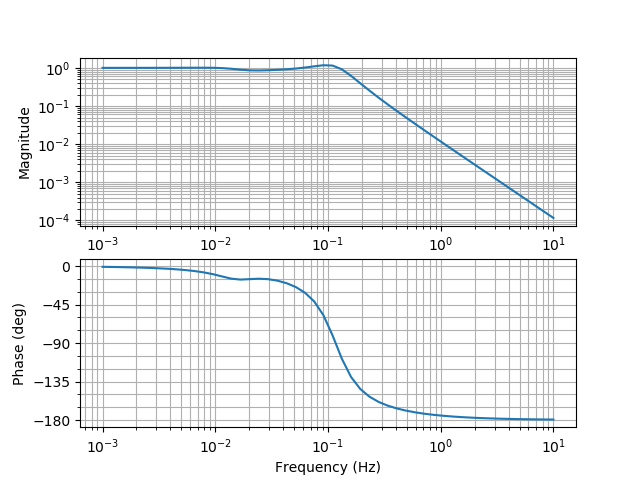
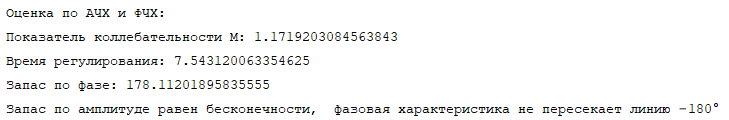


Рисунок 4 - ЛАЧХ и ЛФЧХ системы САУ.

Вывод программы:



1. **Интегральный метод**

Интегральные оценки качества представляют собой интегралы по времени (в пределах от 0 до ) от некоторой функции управляемой переменной или сигнала ошибки . Интегральные оценки не связаны непосредственно с количественными оценками переходных процессов. Они чаще всего применяются в качестве функционала оптимизации по быстродействию систем автоматического управления. Простейшей интегральной оценкой является линейная интегральная оценка, которая равна площади, заключенной между прямой X(∞) и кривой X(t).

Вывод программы:



1. **Анализ параметров ПИ-регулятора**
   1. **Прямой метод**

На рисунке 1 представлена структурная схема САУ. Для данной схемы на рисунке 2 представлена переходная характеристика.



Рисунок 1. Структурная схема АСР

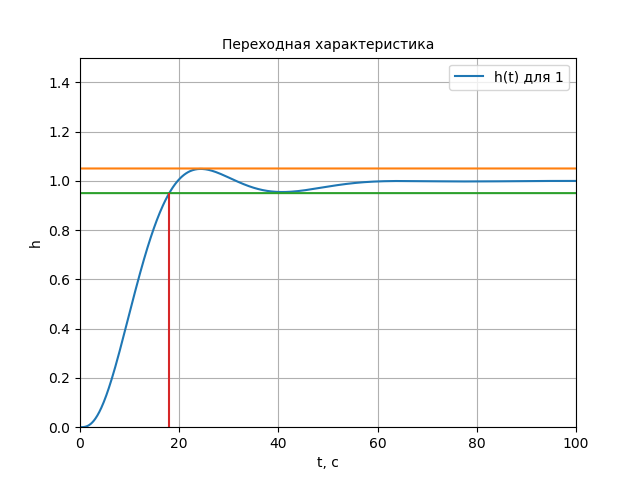
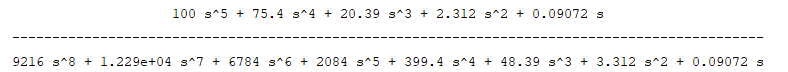
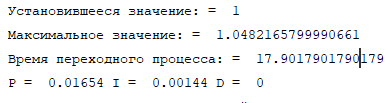


Рисунок 2. Переходная характеристика.

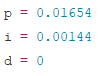
Передаточная функция:



Вывод программы:



Параметры ПИ-регулятора:



* 1. **Корневой метод**

На рисунке 3 представлено распределение корней характеристического уравнения на координатной плоскости:

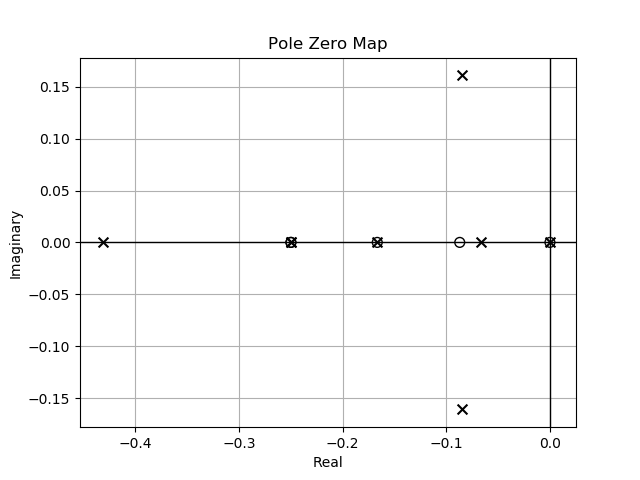
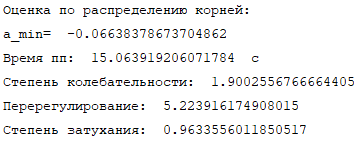


Рисунок 3- Корни характеристического уравнения.

Вывод программы:



**с. По логарифмическим частотным характеристикам**

На рисунке 4 представлены частотные характеристики для данной САУ (ЛАЧХ и ЛФЧХ).

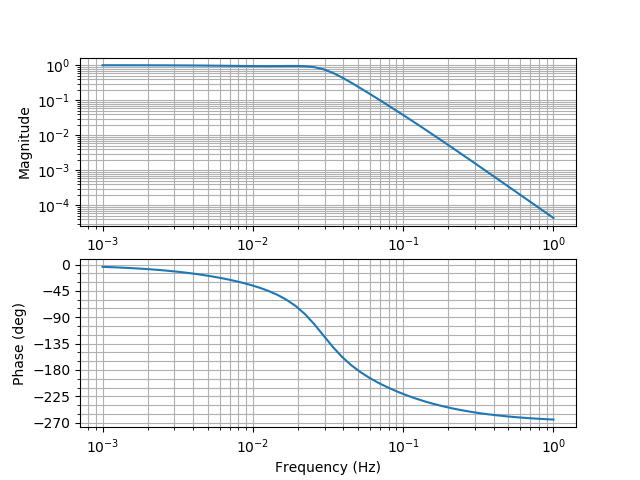
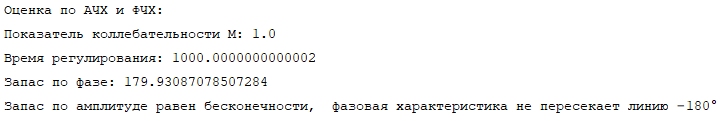
****

Рисунок 4 - ЛАЧХ и ЛФЧХ системы САУ.

Вывод программы:



1. **Интегральный метод**

Интегральные оценки качества представляют собой интегралы по времени (в пределах от 0 до ) от некоторой функции управляемой переменной или сигнала ошибки . Интегральные оценки не связаны непосредственно с количественными оценками переходных процессов. Они чаще всего применяются в качестве функционала оптимизации по быстродействию систем автоматического управления. Простейшей интегральной оценкой является линейная интегральная оценка, которая равна площади, заключенной между прямой X(∞) и кривой X(t).

Вывод программы:



# Вывод:

В данной работе был проведен анализ степени влияния параметров регулятора на устойчивость САУ и подбор данных параметров методом ручного подбора.

Сравнивая результаты определения параметров качества системы, можно прийти к выводу, что корневой метод, как косвенный, отражает иные результаты.

В отношении ПИ-регулятора, методом ручного подбора достаточно сложно подобрать такие коэффициенты, чтобы параметры системы точно удовлетворяли заданным условиям.

# Приложение А: код программы.

import numpy as np  
import matplotlib.pyplot as plt  
import control.matlab as con  
import control as c  
import sympy as sp  
import math  
  
  
def PID(Kp,Ki,Kd):  
 p = con.tf(Kp, 1) + con.tf([0, Ki], [1, 0])  
 return p  
  
def ust(spisok\_y1):  
 state = float(spisok\_y1[len(spisok\_y1) - 1])  
 if (state > 0.95 and state < 1.05):  
 Valueofstate = 1  
 else:  
 Valueofstate = 0  
  
 return Valueofstate  
  
def transition\_time\_of\_proсess\_andGraph(y1, state, t1, Graph):  
 c = 0  
 u = 0  
 T = True  
 y = list(y1)  
 y.reverse()  
 t = list(t1)  
 t.reverse()  
 line1 = 1.05 \* state  
 line2 = 0.95 \* state  
 while (T and c < len(y)):  
 if (float(y[c]) < line1) and (float(y[c]) > line2):  
 c += 1  
 else:  
 T = False  
 transition\_time = t[c]  
 if (not T) and Graph:  
 # y.reverse()  
 line11 = [line1 for i in range(len(y1))]  
 line22 = [line2 for i in range(len(y1))]  
 line3 = np.arange(0, y[c], 0.001)  
 c = np.ones((1, len(line3))) \* t[c]  
 spisok\_time = c[0]  
 lines = [y, line11, line22, line3]  
 lines[0], lines[1], lines[2], lines[3] = plt.plot(t, y, t, line11, t, line22, spisok\_time, line3)  
 plt.legend(lines, [**'h(t) для 1'**], loc=**'best'**, fontsize=10)  
 plt.title(**'Переходная характеристика'**, fontsize=10)  
 plt.ylabel(**'h'**)  
 plt.xlabel(**'t, c'**)  
 plt.axis([0, 100, 0, 1.5])  
 plt.grid()  
 plt.show()  
 return transition\_time  
  
def Pole(W):  
 con.pzmap(W)  
 plt.plot()  
 plt.grid(True)  
 plt.show()  
 print(**'Оценка по распределению корней:'**)  
 Pol = con.pole(W)  
 # print(Pol)  
 P = []

**""" Показатель колебательности характеризует склонность системы к  
 колебаниям: чем выше М, тем менее качественна система при прочих  
 равных условиях. Считается допустимым, если 1,1 < М < 1,5. """** degreeovershoot\_M = []  
 for i in Pol:  
 k = complex(i)  
 if k.real != 0:  
 P.append(k.real)  
 m = k.imag / k.real  
 degreeovershoot\_M.append(m)  
 a\_min = max(P)  
 t\_reg = abs(1 / a\_min)  
 overshooting = math.exp(math.pi / max(degreeovershoot\_M))  
 psi = 1 - math.exp((-2) \* math.pi / max(degreeovershoot\_M))  
 print(**"a\_min= "**, a\_min)  
 print(**"Время пп: "**, t\_reg, **" c"**)  
 print(**"Степень колебательности: "**, max(degreeovershoot\_M))  
 print(**"Перерегулирование: "**, overshooting)  
 print(**"Степень затухания: "**, psi)  
  
def Freq(W):  
 print(**"Оценка по АЧХ и ФЧХ:"**)  
 # t = np.linspace(0, stop=100, num=1000)  
 mag, phase, omega = con.bode(W, dB=False)  
 plt.plot()  
 # plt.close()  
 plt.show()  
 mag\_max = max(mag)  
 M = (mag\_max / mag[0])  
 print(**"Показатель коллебательности М: "** + str(M))  
 t = True  
 n = list(mag).index(mag\_max)  
 while n < len(mag) and t:  
 if ((mag[n] > mag[0] - 0.01) and (mag[n] < mag[0] + 0.01)) or mag[n] < mag[0]:  
 wc = omega[n]  
 phase\_res = phase[n]  
 print(**"Время регулирования: "** + str(2 \* math.pi / wc))  
 **""" В хорошо демпфированных системах запас устойчивости по амплитуде колеблется   
 в пределах от 6 до 20 дБ, а запас по фазе от 30 до 60"""** print(**"Запас по фазе: "** + str(180 - abs(phase\_res)))  
 t = False  
 n = n + 1  
  
 t\_magn = True  
 n = 0  
 while n < len(phase) and t\_magn:  
 if -179.7 > phase[n] > -180.7:  
 index\_phase\_180 = list(phase).index(phase[n])  
 print(**"Запас по амплитуде: "** + str(round(mag[0] - mag[index\_phase\_180], 3)))  
 t\_magn = False  
 n = n + 1  
 **"""запас по амплитуде может быть равен бесконечности, если фазовая характеристика не   
 пересекает линию −180°"""** if t\_magn:  
 print(**"Запас по амплитуде равен бесконечности, фазовая характеристика не пересекает линию −180°"**)  
  
def intergalnaya\_Otsenka(W):  
 a = 0  
 b = 100  
 n = 1000  
 h = (b - a) / n  
 t = np.linspace(a, b, num=n)  
 y, x = con.step(W, t) # х-время ПП  
 func = 0  
 x0 = x[0]  
 y0 = y[0]  
 # нахождение площади методом трапеций  
 for i in range(0, len(x), 1):  
 xi = x[i]  
 y1 = y[i]  
 func += abs(1 \* (xi - x0) - 0.5 \* (y1 + y0) \* (xi - x0))  
 x0 = xi  
 y0 = y1  
 # print(func)  
 print(**"Интегральная оценка за "**, b, **" c = "**, func)  
  
t = np.linspace(0, stop=1000, num=10000)  
treg = 13 # Время регулирования  
overshoot = 1.25  
  
G0 = con.tf(1, 1) # обратная связь  
G1 = con.tf(1, [4, 1]) # generator  
G2 = con.tf(3, [6, 1]) # turbine  
G3 = con.tf(21, [4, 1]) # device  
  
p = 0.01654  
i = 0.00144  
d = 0  
  
Greg = PID(p,i,d)  
G5 = G1 \* G2 \* G3 \* Greg  
G\_zam = G5 / (1 + G5)  
y1, t = con.step(G\_zam,t)  
Value\_state = ust(y1)  
maximom\_of\_func = max(y1)  
time = transition\_time\_of\_proсess\_andGraph(y1, Value\_state, t, False)  
print(G\_zam)  
print(**"Установившееся значение: = "**, Value\_state)  
print(**"Максимальное значение: = "**, maximom\_of\_func)  
print(**"Время переходного процесса: = "**, time)  
print(**"P = "**, p, **"I = "**, i, **"D = "**, d)  
transition\_time\_of\_proсess\_andGraph(y1, Value\_state, t, True)  
Pole(G\_zam)  
Freq(G\_zam)  
intergalnaya\_Otsenka(G\_zam)