Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение   
высшего образования

**Иркутский национальный исследовательский технический университет**

|  |
| --- |
| Институт информационных технологий и анализа данных |
| наименование института |

|  |
| --- |
| **Отчет** |
| по лабораторной работе №1 по дисциплине «Теория оптимального управления»  «РАСЧЕТ НА ЭВМ ОПТИМАЛЬНЫХ ПАРАМЕТРОВ ПИД-РЕГУЛЯТОРА» |
| наименование темы  Вариант № 22 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Выполнил студент |  | АСУб-20-2 |  |  |  | А.В. Арбакова |
|  |  | шифр |  | подпись |  | И.О. Фамилия |
| Проверил |  |  |  |  |  | И.А. Серышева |
|  |  |  |  | подпись |  | И.О. Фамилия |
| Работа защищена с оценкой | | | |  | | |

Иркутск 2024 г.

**Содержание**

[1. Структурная схема автоматической системы с заданной передаточной функцией объекта регулирования; с заданными значениями параметров; заданный вид интегрального критерия оптимизации; вид алгоритма оптимизации. 3](#_Toc158068002)

[2. Изложение выбранного методы моделирования применительно к заданной автоматической системе регулирования. 4](#_Toc158068003)

[3. Фрагмент листинга программы, относящийся к моделированию автоматической системы регулирования, алгоритма параметрической оптимизации. 6](#_Toc158068004)

[4. Вывод выражений, позволяющих вычислить функции чувствительности . 9](#_Toc158068005)

[5. Алгоритм параметрической оптимизации, основанный на градиентной итеративной процедуре. 13](#_Toc158068006)

[6. Картины сходимости, позволяющие сделать выводы о работоспособности алгоритма параметрической оптимизации. 15](#_Toc158068007)

[7. Зависимости c выводом о выполнении необходимого условия оптимальности. 17](#_Toc158068008)

[8. Зависимости позволяющие сделать выводы о работоспособности алгоритма параметрической оптимизации. 19](#_Toc158068009)

[9. Графики переходных процессов h(t) в начальных и конечных точках работы алгоритма параметрической оптимизации. 20](#_Toc158068010)

# **Структурная схема автоматической системы с заданной передаточной функцией объекта регулирования; с заданными значениями параметров; заданный вид интегрального критерия оптимизации; вид алгоритма оптимизации.**

**Цель работы**: ознакомление с методикой расчета оптимальных параметров регулятора при заданной структурной автоматической системы регулирования и заданных параметрах регулирования по интегральным критериям качества процесса регулирования с использованием средств вычислительной техники.

**Вариант**: 22.

**Условия задания** (см. рис. 1).





Рисунок 1 – Заданные значения параметров

**Структурная схема автоматической системы** (см. рис. 2).

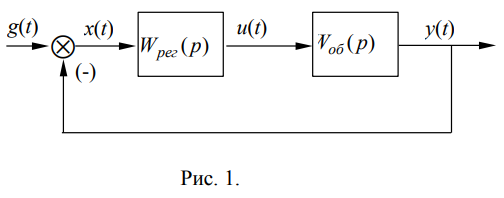


Рисунок 2 – Структурная схема автоматической системы

x(t) – ошибка регулирования.

g(t) – задающее воздействие.

y(t) – выходная величина.

u(t) – регулирующее воздействие.

Wоб(p) – передаточная функция объекта регулирования.

Wрег(p) – передаточная функция регулирующего устройства.

**Передаточная функция регулятора**, представляющая собой пропорционально-интегрально-дифференциальный (ПИД) закон регулирования (см. рис. 3).



Рисунок 3 – Передаточная функция регулятора

q1, q2, q3 – настраиваемые параметры.

**Передаточная функция объекта регулирования** (см. рис. 4).

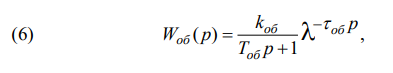


Рисунок 4 – Передаточная функция объекта регулирования

**Критерий оптимизации** (см. рис. 5) – модульный.

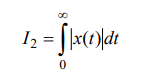


Рисунок 5 – Заданный вид интегрального критерия оптимизации

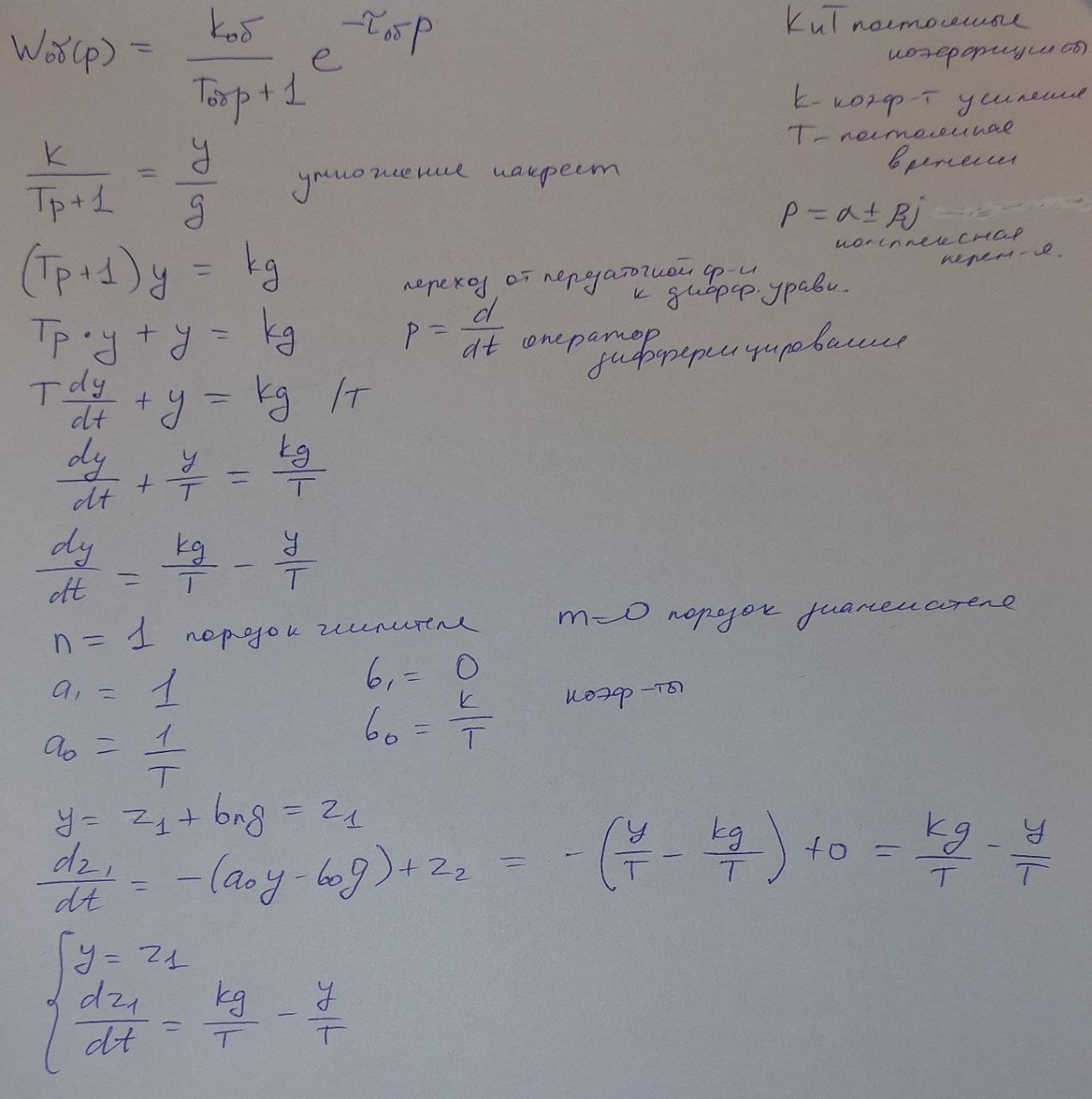
**Вид алгоритма оптимизации***:* Метод релаксации (1).

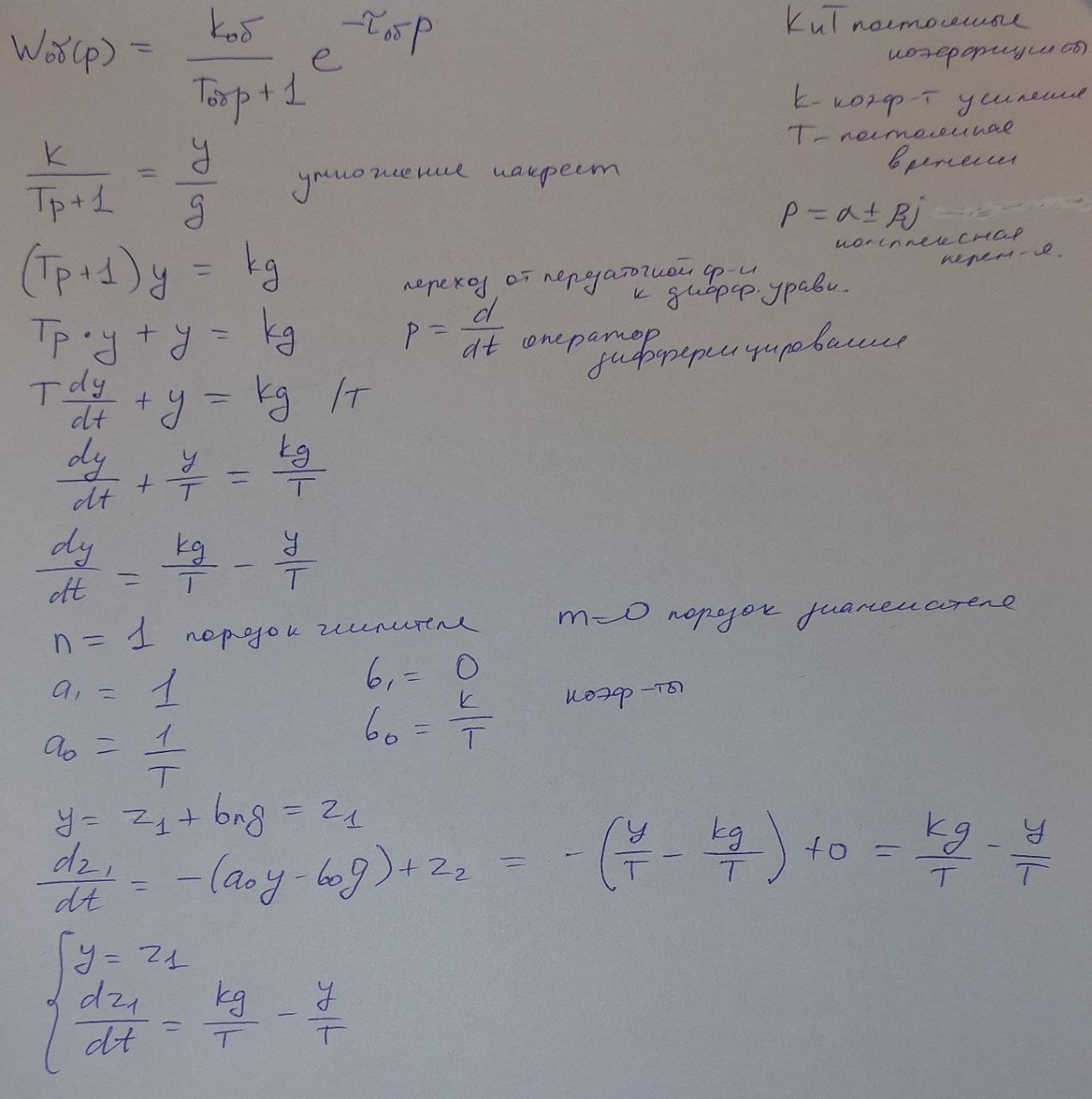
**Задание**:

1. Дифференцируя частным образом по каждому из настраиваемых параметров q1, q2, q3 получить выражения, позволяющие определять функции чувствительности .
2. Записать выражение для каждой из составляющих градиента .
3. Подготовить к моделированию уравнения, описывающие процессы в автоматической системе.

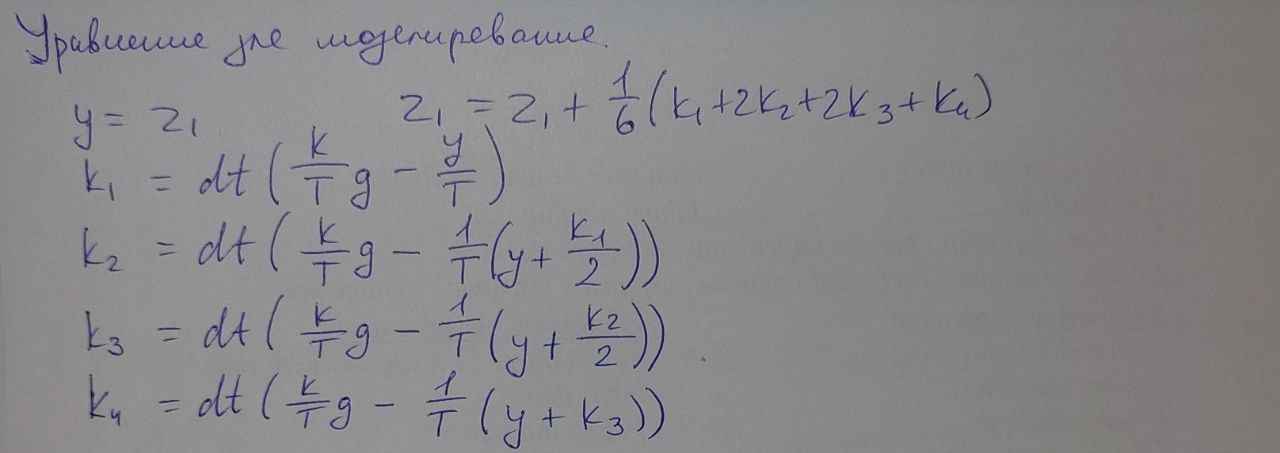
# **Изложение выбранного методы моделирования применительно к заданной автоматической системе регулирования.**

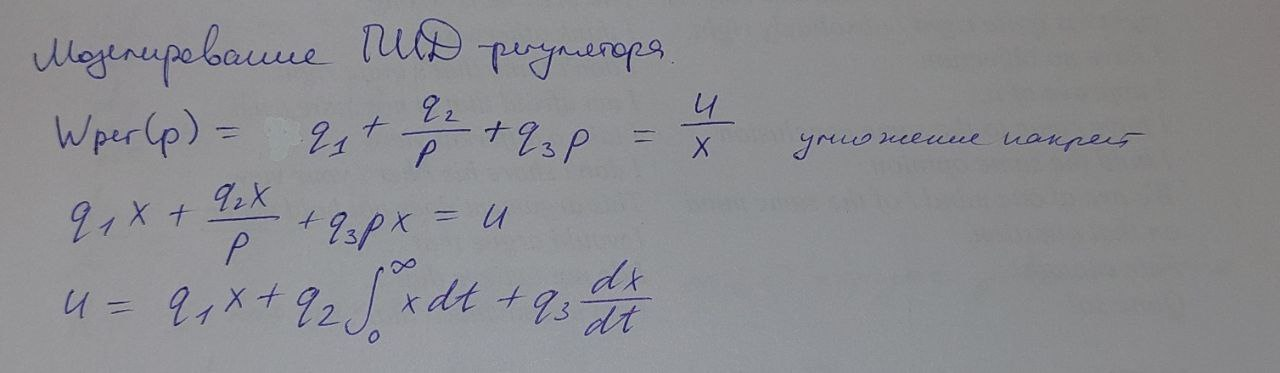
Моделирование передаточной функции объекта производится с помощью метода непосредственного интегрирования.





Для численного решения дифференциального уравнения воспользуемся методом Рунге-Кутта.





# **Фрагмент листинга программы, относящийся к моделированию автоматической системы регулирования, алгоритма параметрической оптимизации.**

Фрагмент листинга программы, описывающий моделирование автоматической системы регулирования:

for (let count = 0; count < 3; count++){

if (count == 0){ //q1, q2, q3 при первом запуске

q1 = 5;

q2 = 2;

q3 = 0.5;

} else if (count == 1){ //q1, q2, q3 при втором запуске

q1 = 1;

q2 = 4;

q3 = 0.2;

} else if (count == 2){ //q1, q2, q3 при третьем запуске

q1 = 2;

q2 = 4.5;

q3 = 0.1;

}

dudq1 = 0, dudq2 = 0, dudq3 = 0;

x = 0;

h = 0.3;

i = 1;

check = true;

for (let kq = 0; kq < Len; kq++){ //цикл для получения значений критерия, q1, q2, q3, составляющих градиента

kritpr = krit; // I\_pred = I

krit = 0; // I = 0

sumx = 0; xpred = 0;

sume1 = 0; sume2 = 0; sume3 = 0;

e1pr = 0; e2pr = 0; e3pr = 0;

sumI = 0; didq1 = 0; didq2 = 0; didq3 = 0;

e1 = 0; e2 = 0; e3 = 0;

y = 0; y1 = 0; //y2=0;y3=0;

z = 0, z1 = 0; z2 = 0; z3 = 0;

t\_ = 0;

i = 1;

j = 0;

while(t\_ <= L ){ //цикл для моделирования переходного процесса, функций чувствительности

xpred = x;

x = g - y;

//u\_pid

sumx = sumx + x \* dt;

dx = (x - xpred) / dt;

u = q1 \* x + q2 \* sumx + q3 \* dx;

//y\_model

sume1 = sume1 + e1 \* dt;

de1 = (e1 - e1pr) / dt;

dudq1 = x - q1 \* e1 - q2 \* sume1 - q3 \* de1;

e1pr = e1;

a = z1;

k1 = dt\*(k / T1 \* dudq1 - (a / (T1)));

k2 = dt\*(k / T1 \* dudq1 - (a + k1 / 2)/T1);

k3 = dt\*(k / T1 \* dudq1 - (a + k2 / 2)/T1);

k4 = dt\*(k / T1 \* dudq1 - (a + k3)/T1);

z1 = z1 + (k1 + 2\*k2 + 2\*k3 + k4)/6;

sume2 = sume2 + e2 \* dt;

de2 = (e2 - e2pr) / dt;

dudq2 = sumx - q1 \* e2 - q2 \* sume2 - q3 \* de2;

e2pr = e2;

b = z2;

k1 = dt\*(k / (T1) \* dudq2 - (b / (T1)));

k2 = dt\*(k / (T1) \* dudq2 - (b + k1 / 2)/T1);

k3 = dt\*(k / (T1) \* dudq2 - (b + k2 / 2)/T1);

k4 = dt\*(k / (T1) \* dudq2 - (b + k3)/T1);

z2 = z2 + (k1 + 2\*k2 + 2\*k3 + k4)/6;

sume3 = sume3 + e3 \* dt;

de3 = (e3 - e3pr) / dt;

dudq3 = dx - q1 \* e3 - q2 \* sume3 - q3 \* de3;

e3pr = e3;

c = z3;

k1 = dt\*(k / (T1) \* dudq3 - (c / (T1)));

k2 = dt\*(k / (T1) \* dudq3 - (c + k1 / 2)/T1);

k3 = dt\*(k / (T1) \* dudq3 - (c + k2 / 2)/T1);

k4 = dt\*(k / (T1) \* dudq3 - (c + k3)/T1);

z3 = z3 + (k1 + 2\*k2 + 2\*k3 + k4)/6;

y1 = z;

m1 = dt\*((k / T1) \* u - (y1 /T1));

m2 = dt\*(k / T1 \* u - (y1 + m1 / 2)/T1);

m3 = dt\*(k / T1 \* u - (y1 + m2 / 2)/T1);

m4 = dt\*(k / T1 \* u - (y1 + m3)/T1);

z = z + (m1 + 2\*m2 + 2\*m3 + m4)/6;

//zap

if(i - 1 > ns) {

if(j >= ns) j = 0;

j = j + 1;

y = mas[j];

e1 = mase1[j];

e2 = mase2[j];

e3 = mase3[j];

mas[j]=y1;

mase1[j] = a;

mase2[j] = b;

mase3[j] = c;

} else {

mas[i]=y1;

mase1[i] = a;

mase2[i] = b;

mase3[i] = c;

e1 = 0;

e2 = 0;

e3 = 0;

y=0;

}

krit = krit + Math.abs(x) \* dt;

didq1 += -Math.sign(x) \* e1 \*dt;

didq2 += -Math.sign(x) \* e2 \*dt;

didq3 += -Math.sign(x) \* e3 \*dt;

if (counter <= L){

A.push([t\_, y, y1, krit]);

B.push([t\_, e1, e2, e3]);

Krit.push([t\_, krit]);

BeforeHt.push(y1);

DT.push(t\_);

}

i++;

t\_ = t\_ + dt;

counter += dt;

if (count == 1){

AfterHt1.push(y1);

}

if (count == 2){

AfterHt2.push(y1);

}

}

if (krit < kritpr)

h = h \* 1.23;

else

h = h / 2;

//ищем максимальный dIdQ

if(Math.abs(didq1) >= Math.abs(didq2) && Math.abs(didq1) >= Math.abs(didq3)){

q1 = q1 - h\*Math.sign(didq1);

} else if(Math.abs(didq2) >= Math.abs(didq1) && Math.abs(didq2) >= Math.abs(didq3)){

q2 = q2 - h\*Math.sign(didq2);

} else {

q3 = q3 - h\*Math.sign(didq3);

}

sumI = didq1 \* didq1 + didq2 \* didq2 + didq3 \* didq3;

if (sumI <= 0.000033){

check = false;

}

if (count == 0){

if (check){

I.push([kq, didq1, didq2, didq3]);

}

Q11.push([q1]);

Q21.push([q2]);

Q31.push([q3]);

KQ.push([kq]);

Q.push([kq,q1,q2,q3]);

Krit1.push([krit]);

} else if(count == 1){

Q12.push([q1]);

Q22.push([q2]);

Q32.push([q3]);

Krit2.push([krit]);

if (check) {

I2.push([kq, didq1, didq2, didq3]);

}

} else if(count == 2){

Q13.push([q1]);

Q23.push([q2]);

Q33.push([q3]);

Krit3.push([krit]);

if (check){

I3.push([kq, didq1, didq2, didq3]);

}

}

}

}

# **Вывод выражений, позволяющих вычислить функции чувствительности .**

Функция чувствительности – частные производные по вариации j-ого параметра выходной координаты системы.

Чувствительностью называют некоторый показатель, характеризующий свойство системы изменять режим работы при отклонении того или иного её параметра от номинальной или исходного значения.

В данной работе структура регулятора задана и представляет собой пропорционально-интегрально-дифференциальный закон регулирования (ПИД-закон регулирования).

П – пропорциональная составляющая, которая увеличивает или уменьшает сигнал ошибки в q раз и является статической; вырабатывает выходной сигнал, противодействующий отклонению регулируемой величины от заданного значения, наблюдаемого в данный момент времени. Регулятор вырабатывает управляющее воздействие на объект пропорционально величине ошибки (чем больше ошибка e, тем больше управляющее воздействие).

И – интегральная составляющая необходима для устранения статической ошибки; пропорциональна интегралу по времени от отклонения регулируемой величины; позволяет регулятору со временем учесть статическую ошибку.

Д – дифференциальная составляющая приводит к уменьшению управляющего воздействия при слишком быстром изменении выходной координаты, то есть делает переходный процесс более плавным; реагирует на изменение сигнала с датчика, и чем сильнее происходит это изменение, тем большее значение прибавляется к общей сумме; позволяет компенсировать резкие изменения в системе и при правильной настройке предотвратить сильное перерегулирование и уменьшить раскачку.

Для реализации алгоритма параметрической оптимизации необходимо вычислить значения каждой j-ой составляющий вектор-градиент:

Для вычисления запишем критерий оптимальности в общем виде:

− функция, имеющая частные производные по каждому из настраиваемых параметров qj (j = 1(1) m).

Выражение для частной производной примет вид:

Так как, сложная функция, то:

Учитывая, что функция ошибки:

То дифференцируя по получаем:

– функция чувствительности, которая характеризует влияние j-го параметра на выходной сигнал. Получаем, что для нахождения частных производных критерия по параметрам необходимо вычислять функции чувствительности. И так как ПИД-регулятор имеет три настраиваемых параметра, то j =1,2,3.

Определим функции чувствительности, поэтому из рисунка 2, описывающий структурную схему автоматической системы, следует, что:

Дифференцируя частным образом правую и левую части этого выражения и учитывая независимость от настраиваемых параметров получим:

Если на вход передаточной функции объекта подать частную производную сигнала по j-ому параметру, то на выходе получим функцию чувствительности.

Далее, учитывая из рисунка 2, следующее:

Дифференцируем по j-ому параметру:

Теперь запишем выражения частных производных по каждому параметру, подставляя выражение , и учитывая, что

В итоге, чтобы вычислить функцию чувствительности, необходимо сначала вычислить , по приведенным выше формулам, и подать на вход в передаточную функцию объекта. Тогда на выходе мы получим искомую функцию чувствительности.

Затем полученные функции чувствительности можно использовать для расчет компонентов градиента:

Найдем для критерия вида:

В этом случае, , поэтому выражение для j−ой составляющей вектор-градиента можно представить в виде:

Порядок вычисления функций чувствительности можно представить в виде блок-схемы, представленной на рисунке 6.

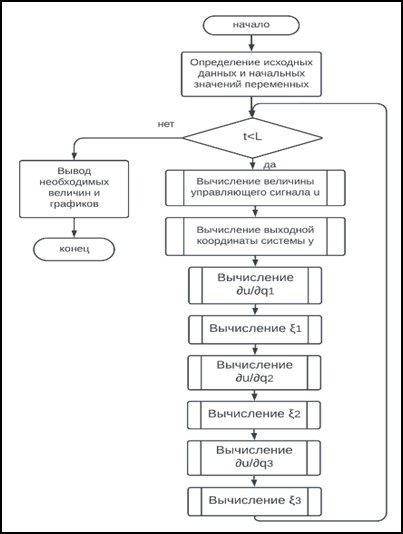


Рисунок 6 – Блок-схема вычисления функций чувствительности

На рисунке 7 представлен график функций чувствительности, для переходной функции без запаздывания h(t), график которой изображен на рисунке 8.

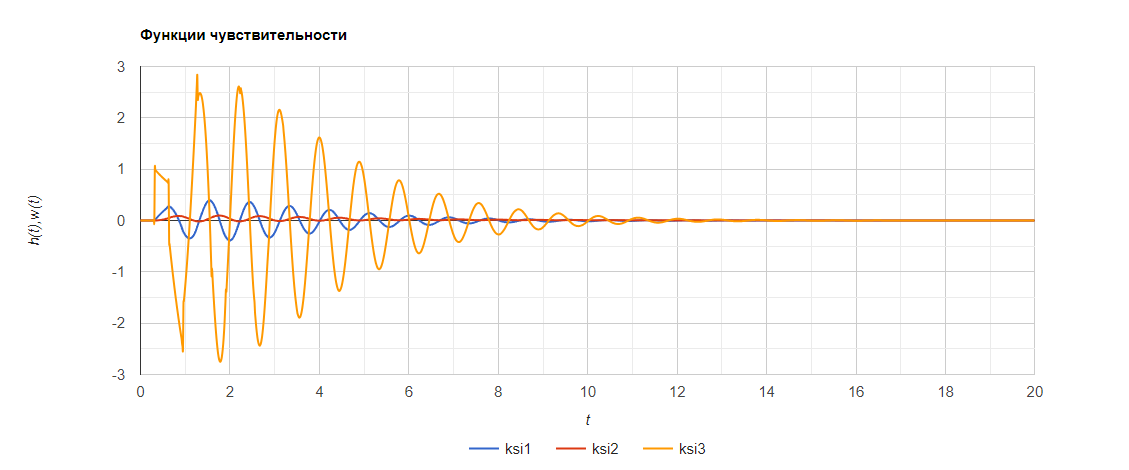


Рисунок 7 – График функций чувствительности

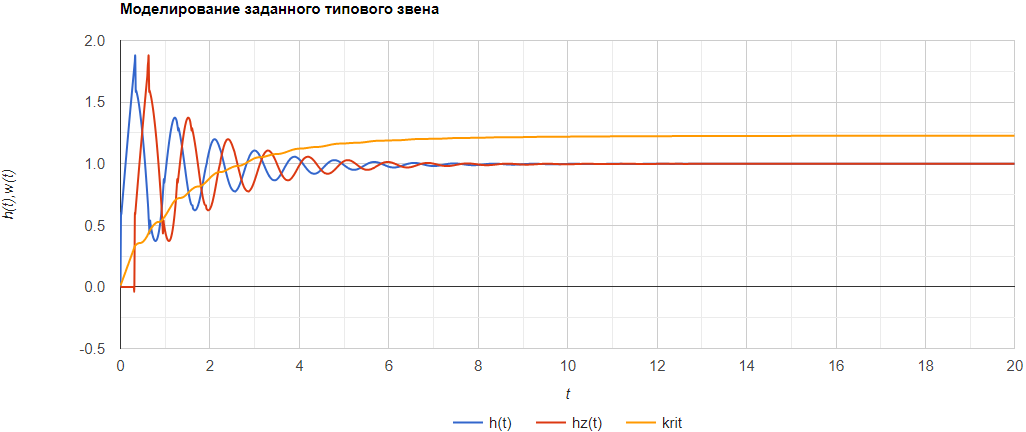


Рисунок 8 – График переходной функции

Зависимость функций чувствительности. Функции чувствительности ищутся для нахождения подходящего критерия, который окажет наилучший эффект на систему. Потом вычисляется градиент и умножается на направление пути, т.е. шаг работы алгоритма.

Если значение функции чувствительности получилось близкое к нулю по 3 критериям одновременно и градиент получился малым, то по критерию остановки алгоритм закончится.

# **Алгоритм параметрической оптимизации, основанный на градиентной итеративной процедуре.**

Метод релаксации – данный метод заключается в том, что необходимо найти направление, вдоль которого критерий уменьшается наиболее сильно. Тем самым, определяется наибольшая по модулю составляющая градиента и делается шаг в эту сторону.

Изменяться параметры по выбранному направлению будут по следующей формуле:

− значение настраиваемого параметра на l-ом шаге работы алгоритма оптимизации.

sign − функция знака.

– шаг работы алгоритма.

– величина -го шага, которая будет изменяться по следующему алгоритму:

Изначально

Графическое изображение движения алгоритма оптимизации от начального значения вектора настраиваемых параметров q[0] к оптимуму q\* в случае двух настраиваемых параметров показано на рисунке 9.

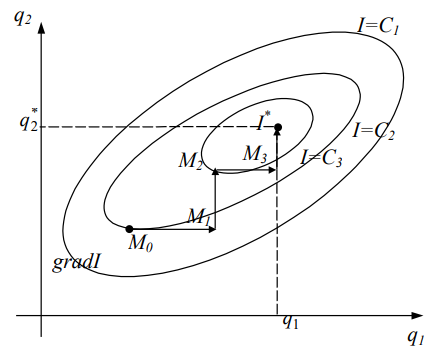


Рисунок 9 – Графическое изображение движения алгоритма оптимизации

На рисунке 10 представлена блок-схема алгоритма.

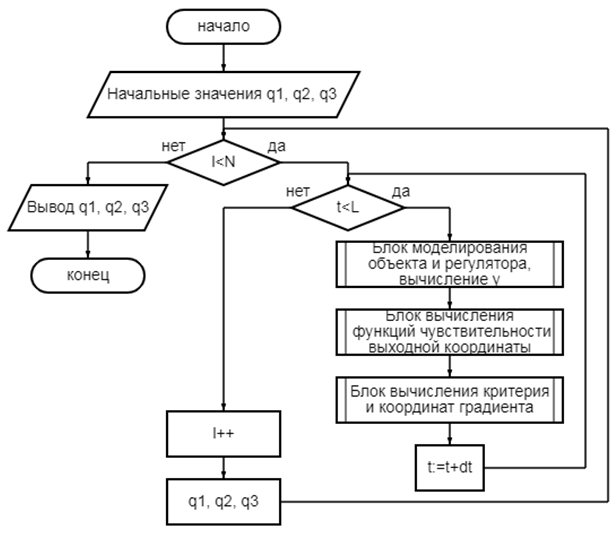


Рисунок 10 – Блок-схема алгоритма

Опишем алгоритм метода:

1. Критерий остановки выполнился? Если да, алгоритм оптимизации останавливается, иначе к 1.1.
   1. Моделируем систему, при этом в процессе моделирования:
      1. Считаем функции чувствительности.
      2. Считаем интеграл:
      3. Считаем компоненты градиента:
   2. Если значение критерия уменьшилось, то переходим на 1.4, иначе на 1.3
   3. Изменяем шаг: . Переходим на 1.5
   4. Изменяем шаг:
   5. Находим максимум компоненты градиента:
   6. Изменяем значение параметра, соответствующее максимальной компоненте:
   7. Возвращаемся к пункту 1.

# **Картины сходимости, позволяющие сделать выводы о работоспособности алгоритма параметрической оптимизации.**

Значения q1 при разных запусках на рисунке 11.

|  |  |  |
| --- | --- | --- |
| **Запуск** | **q1** | **q1\*** |
| **1** | 5 | 2,67 |
| **2** | 1 | 2,672 |
| **3** | 2 | 2,716 |

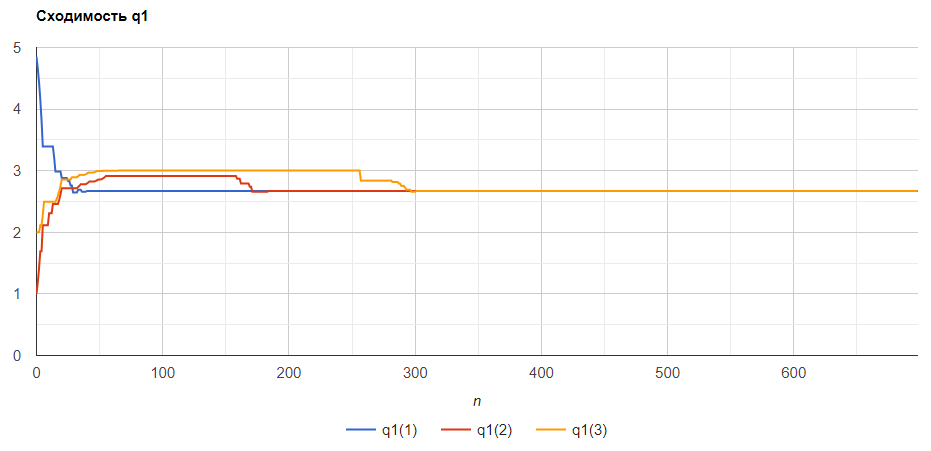


Рисунок 11 – Сходимость q1

Значения q2 при разных запусках на рисунке 12.

|  |  |  |
| --- | --- | --- |
| **Запуск** | **q2** | **q2\*** |
| **1** | 2 | 2,328 |
| **2** | 4 | 2,334 |
| **3** | 4.5 | 2,374 |

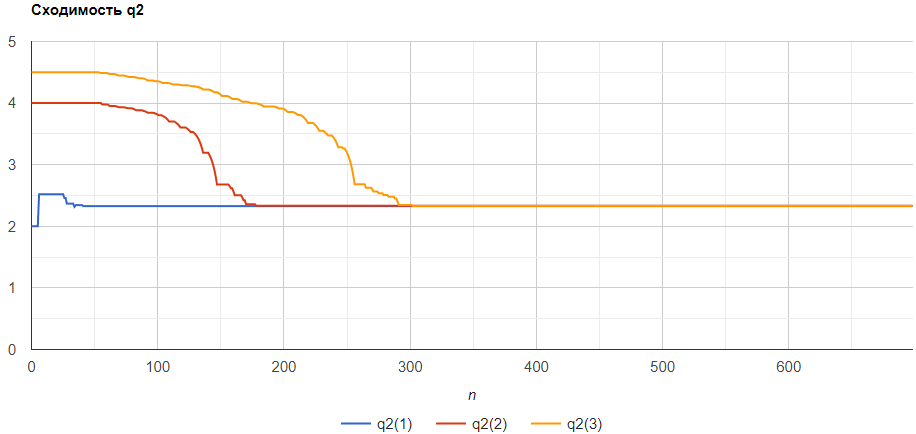


Рисунок 12 – Сходимость q2

Значения q3 при разных запусках на рисунке 13.

|  |  |  |
| --- | --- | --- |
| **Запуск** | **q3** | **q3\*** |
| **1** | 0,5 | 0,375 |
| **2** | 0,2 | 0,376 |
| **3** | 0,1 | 0,389 |

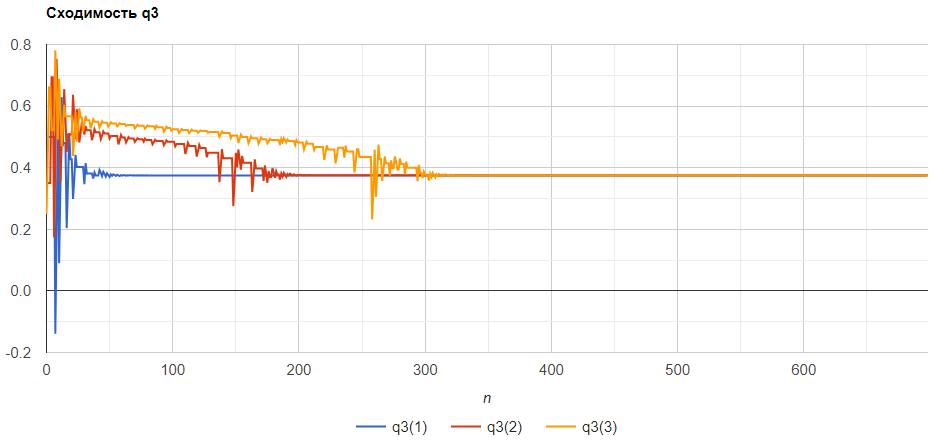


Рисунок 13 – Сходимость q3

На рисунке 14 изображены параметры q1, q2 и q3.

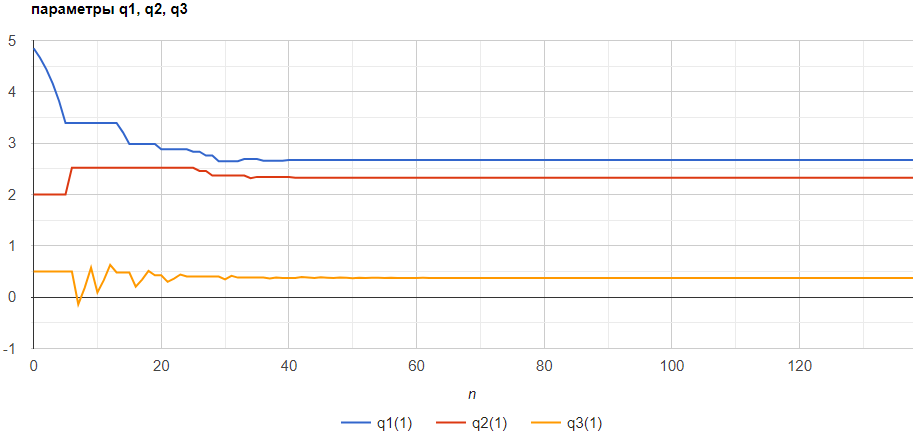


Рисунок 14 – Параметры q1, q2 и q3

q1 = 2,67

q2 = 2,328

q3 = 0,375

Из рисунка 14 видно, что реализован метод релаксации, т.к. в один момент шага изменяется только один параметр.

Из представленных выше примеров можно сделать вывод о том, чтовсе параметры q сходятся примерно к одному значению. Оптимальные значения параметров q сходятся с оптимальными значениями на рисунке 14. Также можно сделать вывод о том, что в качестве алгоритма параметрической оптимизации применялся метод релаксации, т.к. графики имеют ступенчатый вид (параметры q1, q2, q3 пересчитываются только для соответствующей составляющей градиента).

# **Зависимости c выводом о выполнении необходимого условия оптимальности.**

Из представленных ниже графиков (см. рисунок 15, 16 и 17) можно сделать вывод, что необходимое условие оптимальности выполняется (близость составляющих градиента к 0). Это означает, что искомый минимум найден.

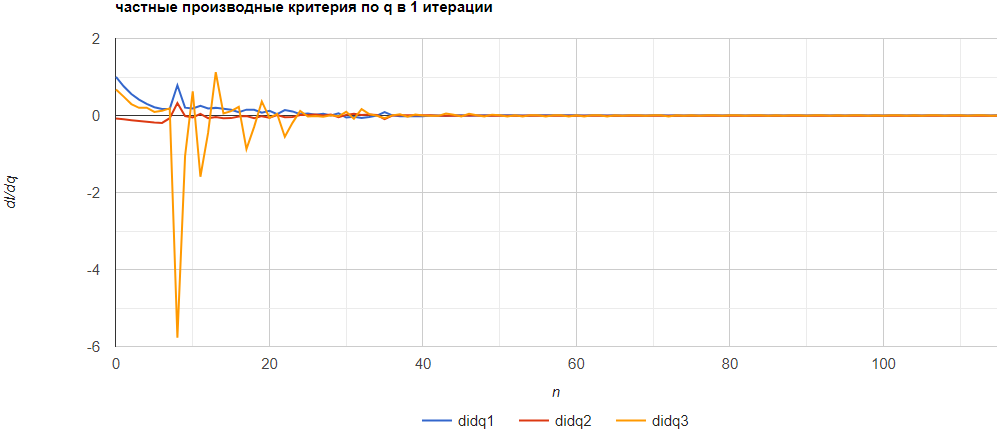


Рисунок 15 – Зависимость от шага алгоритма в первом запуске



Рисунок 16 – Зависимость от шага алгоритма во втором запуске



Рисунок 17 – Зависимость от шага алгоритма в третьем запуске

# **Зависимости позволяющие сделать выводы о работоспособности алгоритма параметрической оптимизации.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Параметр** | **1 запуск** | **2 запуск** | **3 запуск** |
|  | 1,226856 | 1,593196 | 2,428607 |
|  | 0,448506 | 0,448521 | 0,449381 |

Более точные значения параметров I0 (krit10, krit20, krit30) и I (krit1, krit2, krit3) представлены на рисунке 18.

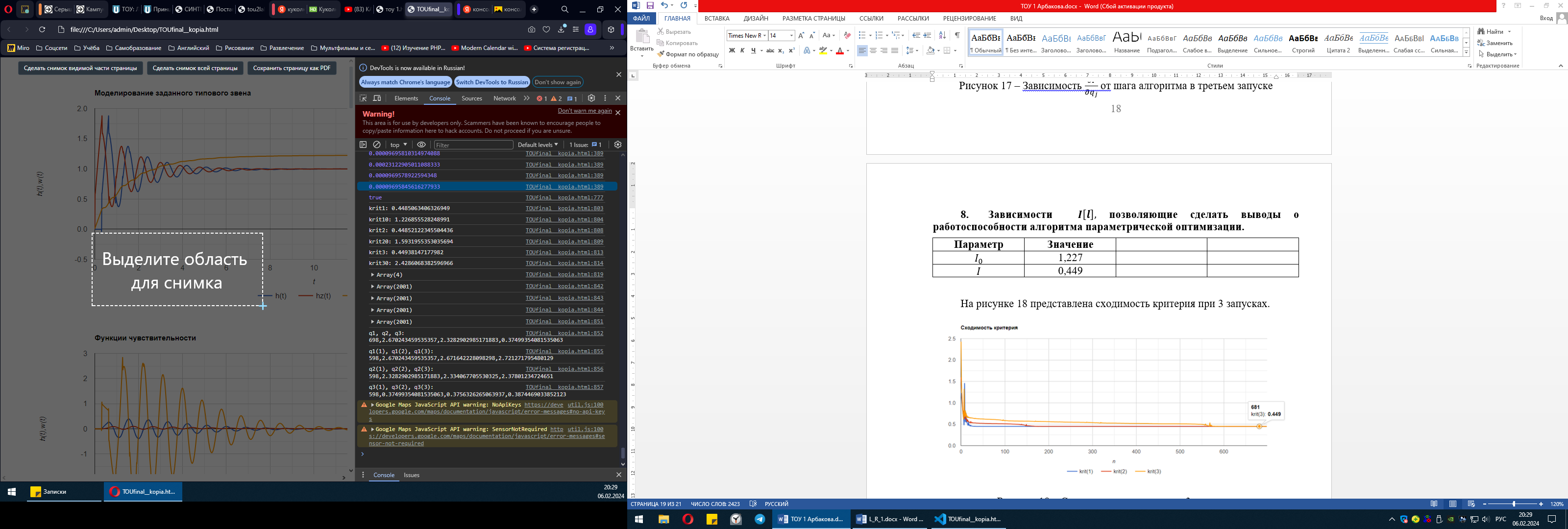


Рисунок 18 – Точные значения параметров I0 и I

На рисунке 19 представлена сходимость критерия при 3 запусках.



Рисунок 19 – Сходимость критерия при 3 запусках

Из представленных выше примеров можно сделать вывод о том, чтов ходе оптимизации критерий I сходится к одному значению при 3 запусках.

Ошибка минимизируется до значения 0,449.

# **Графики переходных процессов h(t) в начальных и конечных точках работы алгоритма параметрической оптимизации.**

На рисунке 20 представлен переходный процесс до оптимизации и после оптимизации.

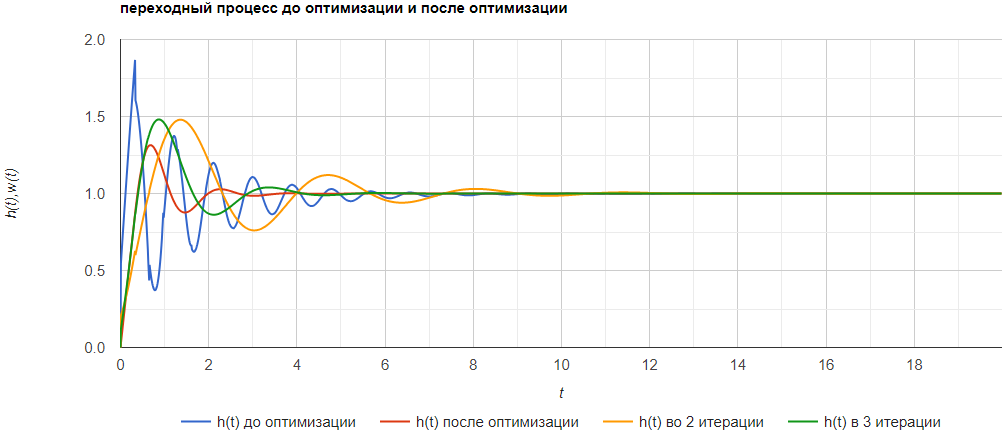


Рисунок 20 – Переходный процесс до оптимизации и после оптимизации

На рисунке 21 представлен переходный процесс до оптимизации и после оптимизации в 1 итерации.

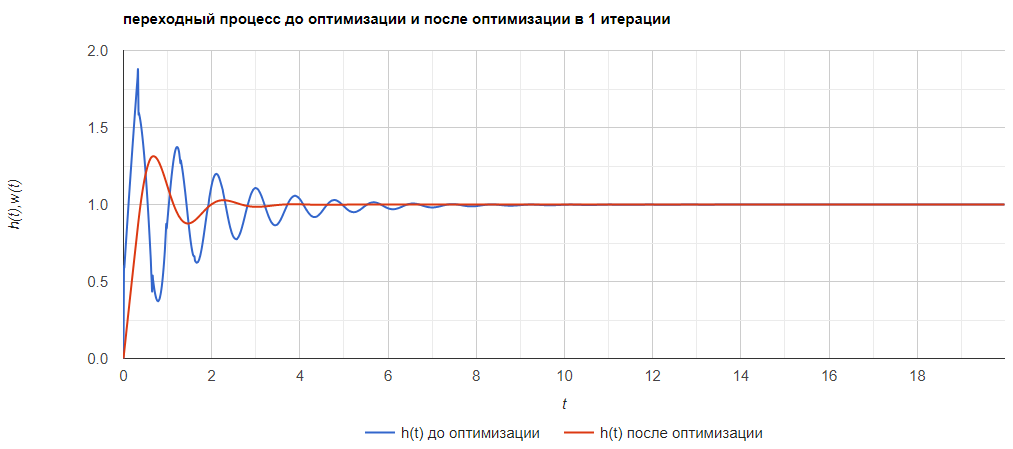


Рисунок 21 – Переходный процесс до оптимизации и после оптимизации в 1 итерации

На рисунке 22 представлен переходный процесс до оптимизации и после оптимизации во 2 итерации.

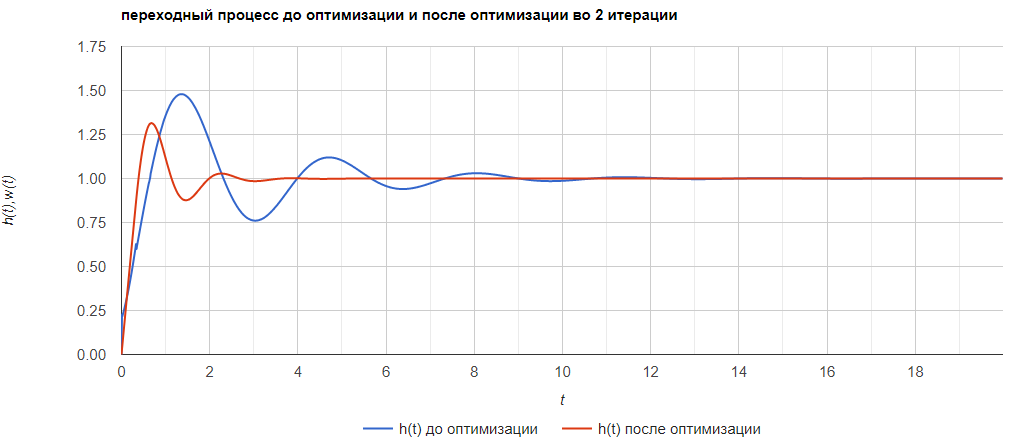


Рисунок 22 – Переходный процесс до оптимизации и после оптимизации во 2 итерации

На рисунке 23 представлен переходный процесс до оптимизации и после оптимизации в 3 итерации.

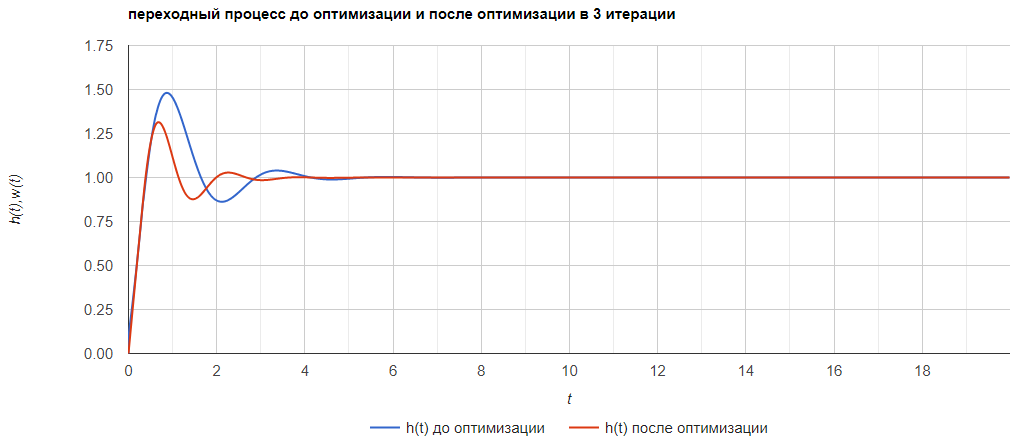


Рисунок 23 – Переходный процесс до оптимизации и после оптимизации в 3 итерации

По рисункам, представленным выше видно, что оптимальный переходный процесс идентичен во всех 3 запусках. При оптимально подобранных параметрах q1, q2, q3 переходный процесс происходит быстрее, чем при первоначальных значениях параметров.