



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

---

ФАКУЛЬТЕТ «Специальное машиностроение»

КАФЕДРА «Колесные машины»

Отчёт о выполнении лабораторной работы №7  
по курсу  
«Управление техническими системами»  
на тему  
«Следящая система управления дроссельной заслонкой  
автомобиля»

Студент СМ10-81

\_\_\_\_\_  
(подпись, дата)

В.Б. Сухоносенко

\_\_\_\_\_  
(Ф.И.О.)

Преподаватель

\_\_\_\_\_  
(подпись, дата)

А.А. Смирнов

\_\_\_\_\_  
(Ф.И.О.)

2025 г.

# СОДЕРЖАНИЕ

|                                                               |    |
|---------------------------------------------------------------|----|
| ВВЕДЕНИЕ . . . . .                                            | 3  |
| 1   Выполнение работы . . . . .                               | 4  |
| 1.1   Тарировка датчиков . . . . .                            | 4  |
| 1.2   Программная реализация регулятора и его настройка . . . | 5  |
| ЗАКЛЮЧЕНИЕ . . . . .                                          | 14 |

## ВВЕДЕНИЕ

Задачей лабораторной работы №6 являлось:

1. Изучение устройства и принципа работы дроссельной заслонки и педали акселератора,
2. Изучение теоретических основ работы системы управления дроссельной заслонкой,
3. Реализация программного ПИД-регулятора для системы управления и его исследование.
4. Подбор коэффициентов ПИД-регулятора,
5. Анализ результатов эксперимента.

# 1 Выполнение работы

## 1.1 Тарировка датчиков

Показания датчиков угла поворота педали акселератора и дроссельной заслонки определены с помощью специально написанной программы для платы Arduino. По последовательному интерфейсу передаются данные о текущем уровне напряжения датчиков.

Ниже приведён листинг данной программы.

```
//VOVKA SUHONOSSENKO IDENTIFIKATSIA
//Объявление переменных
const int PedalSensorPin = A0; //аналоговый вход с потенциометра педали
const int ThrottleSensorPin = A1; //аналоговый вход с потенциометра за

void setup() {
  Serial.begin(9600); //устанавливаем скорость последовательного порта

}

void loop() {
  //оцифровываем напряжение на потенциометре педали
  int PedalValue = analogRead(PedalSensorPin);
  //оцифровываем напряжение на потенциометре заслонки
  int ThrottleValue = analogRead(ThrottleSensorPin);

  Serial.print("Pedal: "); //передаем через последовательный порт
  Serial.print(PedalValue); //значение датчика педали
  Serial.print("\tThrottle: "); //передаем через последовательный порт
  Serial.print(ThrottleValue); //значение датчика заслонки
  Serial.print("\n");
  delay(300);
}
```

В результате эксперимента построена таблица 1.

Таблица 1 — Предельные значения датчиков

| Состояние                  | Имя переменной   | Значение |
|----------------------------|------------------|----------|
| Педаль полностью отпущена  | PedalValueMin    | 151      |
| Педаль полностью нажата    | PedalValueMax    | 834      |
| Заслонка закрыта           | ThrottleValueMin | 213      |
| Заслонка полностью открыта | ThrottleValueMax | 805      |

С помощью АЦП и программы LGraph получен график напряжения во времени для датчика положения педали акселератора и дроссельной заслонки. В ходе эксперимента сначала нажималась и отпускалась педаль, а затем вручную заслонка отводилась в открытое положение. График показан на рисунке 1.1.

```
close all
figure(1)
plot(exp1(:,1),exp1(:,2),'black')
hold on
plot(exp1(:,1),exp1(:,3),'red')
```

С графика сняты значения максимального и минимального напряжения для обработки результатов второго эксперимента.

## 1.2 Программная реализация регулятора и его настройка

Разработана программа для микроконтроллера, реализующая следующую систему управления дроссельной заслонкой. Задающим воздействием является положение педали акселератора (В программе определяется по уровню напряжения датчика положения педали), управляющим - напряжение на электродвигателе постоянного тока, приводящего в движение в дроссельную заслонку. Ошибка вычисляется по обратной связи - управляемым параметром является положение заслонки, определяемое по уровню напряжения на датчике поворота заслонки.

Интеграл ошибки вычисляется по формуле прямоугольников, производная – по формуле для конечных разностей. ПИД-регулятор задан в формулировке  $K_p + K_i/s + K_d s$ .

Ниже приведён листинг программы.

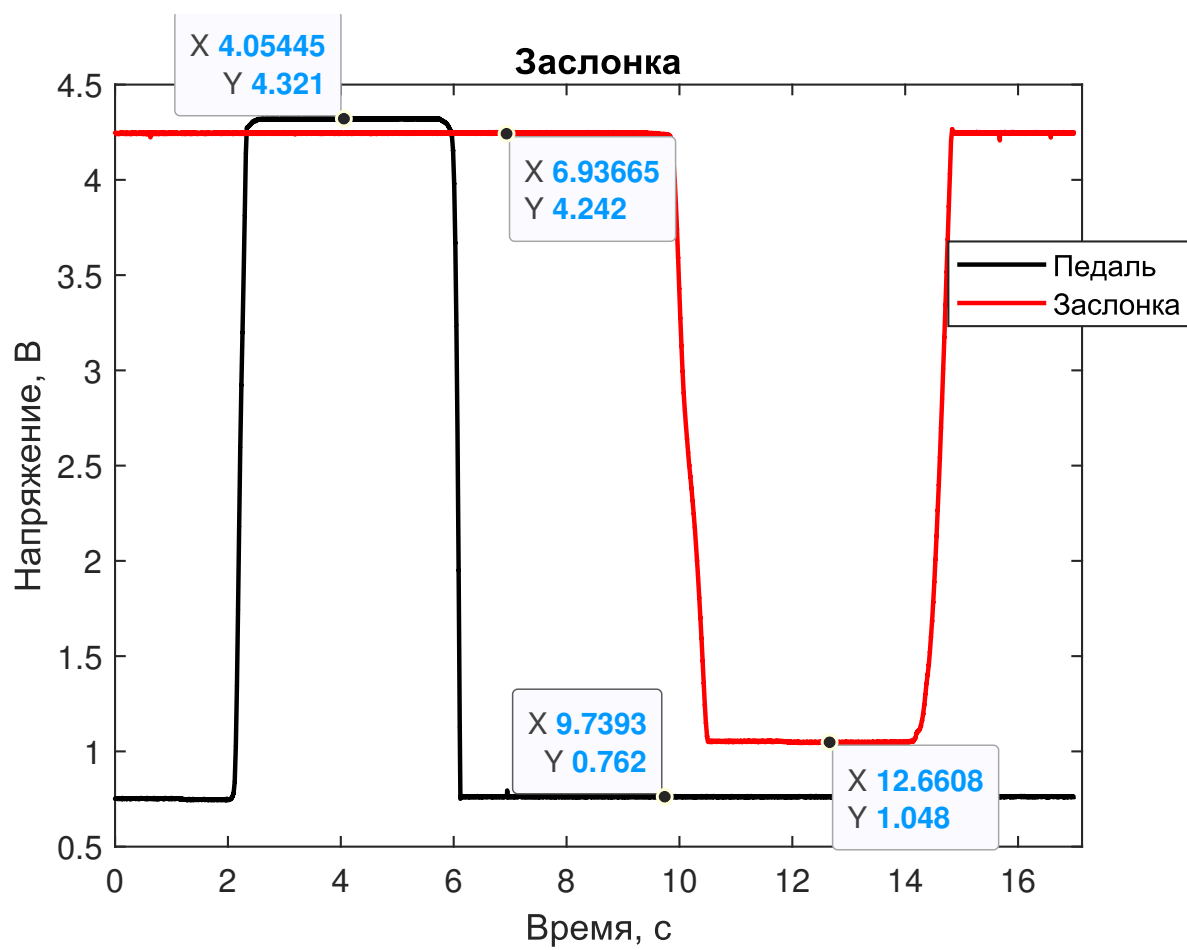


Рисунок 1.1 — Показания датчиков в первом эксперименте

```

//HARRRISON AND SUKHONOSSENKO ORIGINALE
const int PedalSensorPin = A0;
//аналоговый вход с потенциометра педали
const int ThrottleSensorPin = A1;
//аналоговый вход с потенциометра заслонки
const int Potenciometer1 = A3;
//«П» аналоговые входы с потенциометров,
const int Potenciometer2 = A4; //«И» задающих коэффициенты
const int Potenciometer3 = A5; //«Д» ПИД-регулятора
const int MotorThrottlePin = 6;
//выход (с ШИМ) для управл-я мотором заслонки
const float Kpmax = 500;
// Макс. значение коэф. ПИД-регулятора Кп
const float Kimax = 10;
const float Kdmax = 5;
// Макс. значение коэф. ПИД-регулятора Ки
// Макс. значение коэф. ПИД-регулятора Кд
const int PedalValueMin = 151; // см. табл. 1
const int PedalValueMax = 834; //см. табл. 1
const int PedalValueRange = PedalValueMax - PedalValueMin;
//диапазон изменения значений датчика педали
const int ThrottleValueMin = 213; // см. табл. 1
const int ThrottleValueMax = 805; // см. табл. 1
const int ThrottleValueRange = ThrottleValueMax - ThrottleValueMin;
//диапазон изм-я значений датч. заслонки
float eprev = 0.0; /
/значение ошибки на предыдущем шаге работы регулятора
float ie = 0.0; // интеграл ошибки
unsigned long tprev = 0;
//знач-е времени на предыдущ. шаге работы регулятора

void setup() {
Serial.begin(9600);

```

```

//устанавливаем скорость последовательного порта
pinMode(MotorThrottlePin, OUTPUT);
}

//Переменные функции
void loop() {
    unsigned long t = micros(); //текущее время в мкс
    double dt = ((t * 1e-6) - tprev); //dt в секундах!
    tprev = (t * 1e-6) //tprev + dt*1e-6; // прошедшее время в с
    int PedalValue = analogRead(PedalSensorPin);
    //оцифрованное напряжение на потенциометре педали
    int ThrottleValue; //оцифрованное напряжение на потенциометре заслонки
    float g; //задающее воздействие(угол педали)в безразмерном виде
    float y; //выходная переменная (угол заслонки) в безразмерном виде
    float e; //текущая ошибка
    float de_dt; // производная от ошибки de/dt
    float u; //сигнал управления
    int P1value;
    //оцифрованное напряжение на потенциометре 1 (0 ... 1023)
    int P2value;
    //оцифрованное напряжение на потенциометре 2 (0 ... 1023)
    int P3value;
    //оцифрованное напряжение на потенциометре 3 (0 ... 1023)
    float Kp; //коэффициент Kp ПИД-регулятора
    float Ki; //коэффициент Ki ПИД-регулятора
    float Kd; //коэффициент Kd ПИД-регулятора
    g = float((PedalValue - PedalValueMin))/PedalValueRange;
    ThrottleValue = analogRead(ThrottleSensorPin);
    y = float((ThrottleValue - ThrottleValueMin))/ThrottleValueRange;
    e = g - y;

    ie = ie + e*dt;

```



```

de_dt = (e - eprev)/dt;
P1value = analogRead(Potenciometer1);
P2value = analogRead(Potenciometer2);
P3value = analogRead(Potenciometer3);

Kp = P1value*Kpmax/1023;
Ki = P2value*Kimax/1023;
Kd = P3value*Kdmax/1023; // KP KI KD

Kp= 100.59;
Ki=35.36;
Kd=1.90;
// ЭТИ КОЭФФИЦИЕНТЫ ВВЕДЕНЫ ПОСЛЕ ИХ ОПРЕДЕЛЕНИЯ.

u = Kp * e + Ki * ie + Kd * de_dt;
eprev = e;
if (u<0) u = 0;
else if (u>255) u = 255;
analogWrite(MotorThrottlePin, int(u));

Serial.print("g: ");Serial.print(g);
Serial.print(" y: ");Serial.print(y);
Serial.print(" e: ");Serial.print(e);
Serial.print(" u: ");Serial.print(u);
Serial.print(" Kp: ");Serial.print(Kp);
Serial.print(" Ki: ");Serial.print(Ki);
Serial.print(" Kd: ");Serial.print(Kd);
Serial.print("\n");
delay(300);
}

```

Подбор коэффициентов ПИД-регулятора осуществлялся изменением сопротивления трёх потенциометров, показания с которых конвертируются

в коэффициенты ПИД. После определения оптимальных коэффициентов они были зафиксированы в программе и был проведён второй эксперимент.

Подобранные коэффициенты регулятора:

$$K_p = 100.59, K_i = 35.36, K_d = 1.90$$

Эксперимент проводился для следующих воздействий: первое воздействие - ступенчатое – нажатие на педаль до конца, удержание в течение 5 секунд и её отпускание. Второе воздействие - резкое нажатие на педаль и её отпускание; третье воздействие - нажатие педали наполовину и отпускание её через 5 секунд.

С помощью программы LGraph и АЦП считаны значения напряжений с датчиков; построены графики напряжения в MATLAB (рисунок 1.2):

% минимальные и максимальные значения напряжения на датчиках

pedal\_min=0.759

pedal\_max=4.319

throt\_min=1.048

throt\_max=4.245

i=step;

stepn=[i(:,1)

(i(:,2)-pedal\_min)/(pedal\_max-pedal\_min)

1-(i(:,3)-throt\_min)/(throt\_max-throt\_min)]

i=halfstep;

halfstepn=[i(:,1)

(i(:,2)-pedal\_min)/(pedal\_max-pedal\_min)

1-(i(:,3)-throt\_min)/(throt\_max-throt\_min)]

i=impulse;

impulsen=[i(:,1)

(i(:,2)-pedal\_min)/(pedal\_max-pedal\_min)

1-(i(:,3)-throt\_min)/(throt\_max-throt\_min)]

figure(2)

```

tiledlayout(1,3)
nexttile([1 1])
plot(stepn(:,1),stepn(:,2),'black')
hold on
plot(stepn(:,1),stepn(:,3),'red')
xlim([1 8])

nexttile([1 1])
plot(impulsen(:,1),impulsen(:,2),'black')
hold on
plot(impulsen(:,1),impulsen(:,3),'red')
xlim([2.5 4.5])

nexttile([1 1])
plot(halfstepn(:,1),halfstepn(:,2),'black')
hold on
plot(halfstepn(:,1),halfstepn(:,3),'red')

xlim([3 10])

```

В результате эксперимента заметны следующие явления:

При первом воздействии начальная ошибка нулевая, но после окончания ступенчатого воздействия заслонка не возвращается до конца в исходное положение (остаётся открытой примерно на долю 0.13). Имеется задержка между положением педали и заслонки.

При втором воздействии (быстрое нажатие и опускание) отставание заслонки от педали ещё более значительно. В начальный момент времени заслонка сильно открыта (степень открытия 0.1), хотя педаль отпущена. После воздействия доля открытия заслонки уменьшается до 0.04.

При третьем воздействии (открытие наполовину) имеется значительная статическая погрешность до и во время воздействия, которая пропадает после отпускания педали.

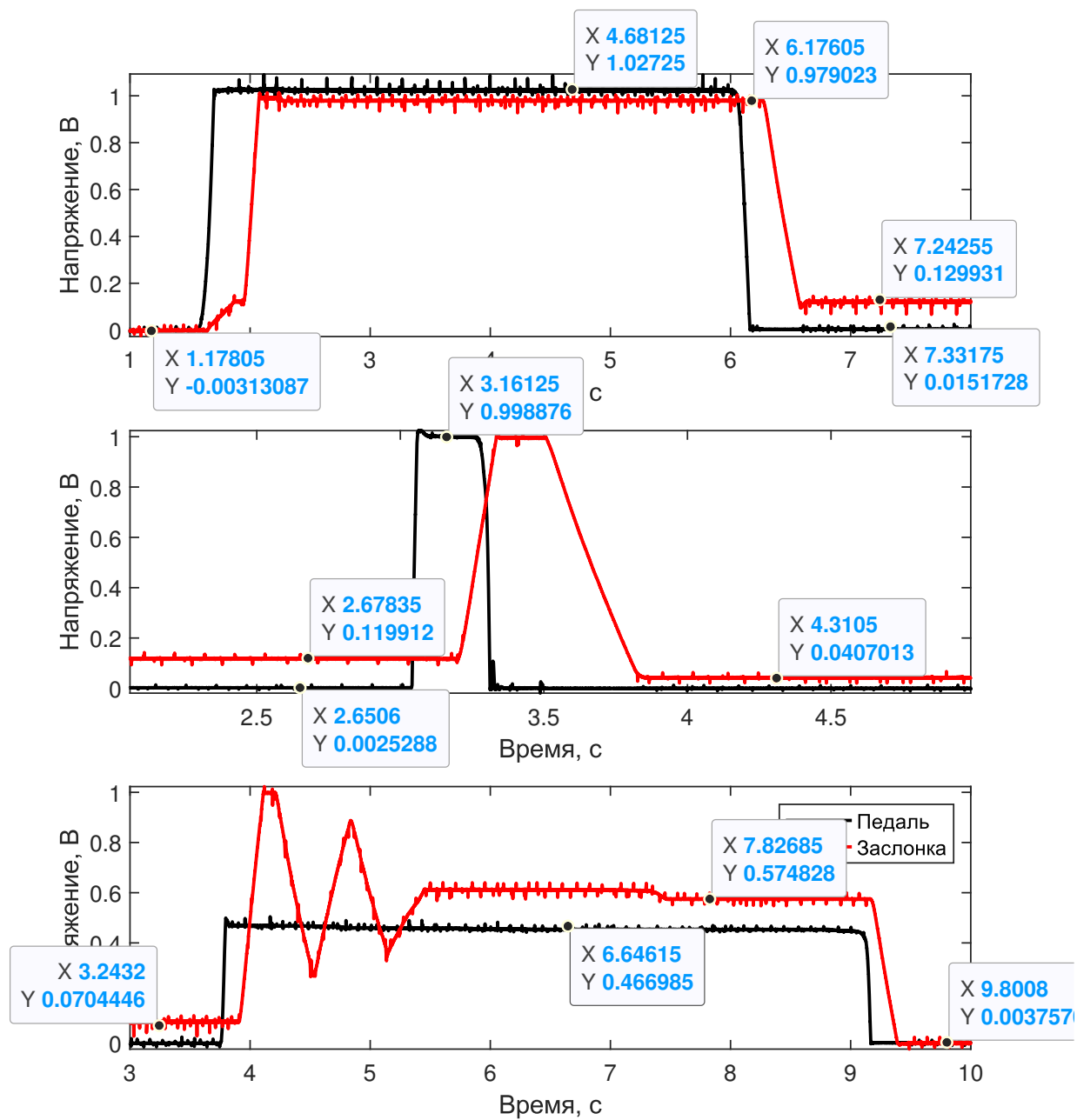


Рисунок 1.2 — Обработанные результаты эксперимента 2

Из перечисленного можно сделать вывод, что подбор коэффициентов ПИД-регулятора можно попробовать улучшить, а так же стоит исследовать причины больших установившихся отклонений, которые можно связать с заеданием заслонки, зависанием программы или неоптимальным вычислением ошибки и действия регулятора и др.

## ЗАКЛЮЧЕНИЕ

В лабораторной работе изучено устройство и принцип работы дроссельной заслонки и педали акселератора, программно реализован ПИД-регулятор для следящей системы управления. Проведён экспериментальный подбор коэффициентов регулятора и выполнен эксперимент с различными задающими воздействиями, после чего результаты были обработаны и проанализированы (рисунок 1.2).

Рассмотрены возможные причины неудовлетворительной работы регулятора при различных задающих воздействиях.