

به نام خدا

## گزارش شماره 9 آزمایشگاه کنترل صنعتی

تهیه کننده: علیرضا امیری

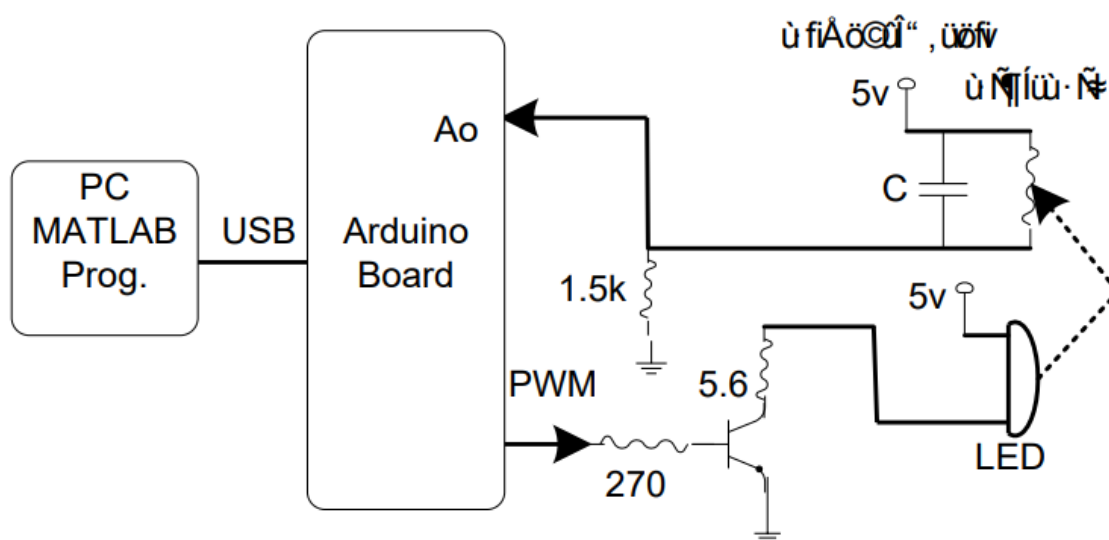
شماره دانشجویی: 982151028

استاد درس: دکتر سیدطبابی

هدف از انجام این آزمایش تعیین مدل یک سیستم لامپ چشمک زدن و سپس کنترل شدت روشنایی آن به وسیله ی یک کنترلر PID می باشد.

### سوال 1-9: تعیین مدل سیستم با پاسخ پله حلقه باز

ابتدا مدار زیر را می بندیم.



این مدار شامل یک لامپ LED است که به صورت متناوب و با دوره تناوب ثابت شدت روشنایی آن به وسیله ی خروجی PWM آردوینو کم و زیاد می شود. از طرفی، با استفاده از یک فتورزیستور که در نزدیکی LED قرار گرفته است، می توانیم شدت روشنایی را اندازه گیری کنیم. بنابراین، می توانیم با داشتن مقادیر خوانده شده از فتورزیستور و با کم و زیاد کردن نور لامپ که نقش پاسخ پله را دارد، تابع تبدیلی برای سیستم تخمین بزنیم.

برای اینجانب مقدار خازن C برابر با 100 میکروفاراد در نظر گرفته شده است. پس از بستن مدار، کد زیر را اجرا می کنیم.

```

int pwm=10;
int sensor=5;
int duty;
int counter;

void setup()
{
  duty=127;
  counter=0;
  Serial.begin(9600);
  pinMode(pwm,OUTPUT);
  pinMode(sensor,INPUT);
}

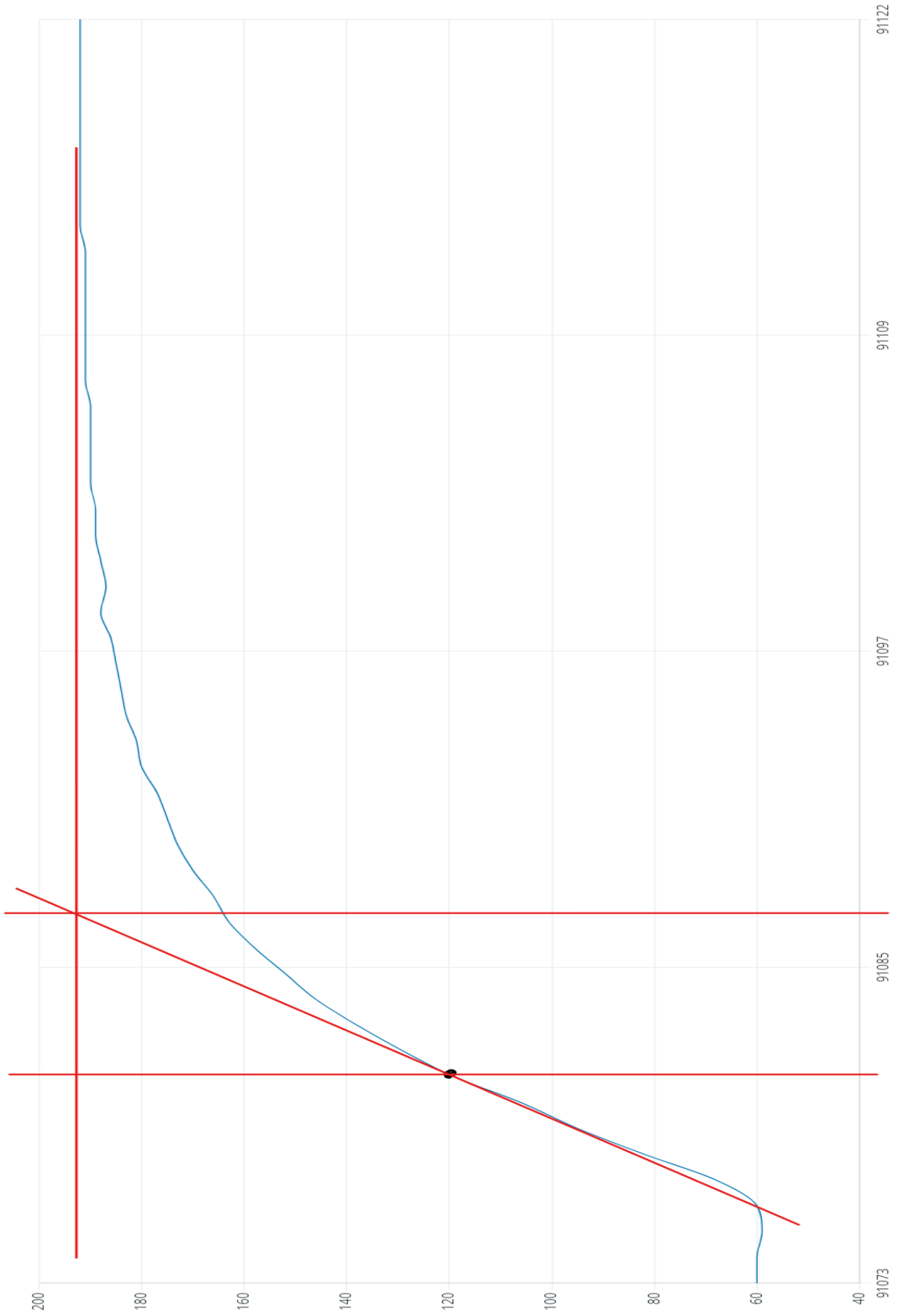
void loop()
{
  if(counter>=200)
  {
    if(duty==10)
    {
      duty=250;
    }
    else
    {
      duty=10;
    }
    counter=0;
  }
  analogWrite(pwm,duty);
  sensor=analogRead(A5);
  sensor=map(sensor,0,1023,0,255);

  Serial.println(sensor);

  counter ++;
}

```

پاسخ پله ی این سیستم که با استفاده از ابزار SerialPlot به دست آمده است، به صورت زیر است:



## سوال 2-9: تعیین مدل سیستم از پاسخ پله

حال باید از نمودار فوق، یک پاسخ پله ی درجه یک تقریبی برای سیستم به دست آوریم. مطابق اندازه گیری های انجام شده، مدت زمان میان هر دو بار نمونه برداری برابر  $5.5ms$  می باشد. همچنین برای اعمال ورودی،  $duty\ cycle$  از مقدار 10 تا 250 تغییر می کند.

(1) تعیین گین  $K$  سیستم:

برای تعیین گین، از رابطه ی زیر استفاده می کنیم:

$$K = \frac{y_{\infty}}{a} = \frac{192 - 60}{250 - 10} = \frac{132}{240} = 0.55$$

(2) تعیین ثابت زمانی سیستم:

برای این منظور، در یک نقطه مماسی بر نمودار رسم می کنیم و سپس از همان نقطه خط عمودی نیز رسم می کنیم. سپس، محل تلاقی خط مماس با خط افقی مقدار نهایی را پیدا می کنیم. حد فاصل خط عمودی و محل تلاقی پیدا شده، برابر با ثابت زمانی است. در سیستم فوق این اندازه گیری ها به شرح زیر است:

$$\tau(samples) = 91087 - 91081 = 6$$

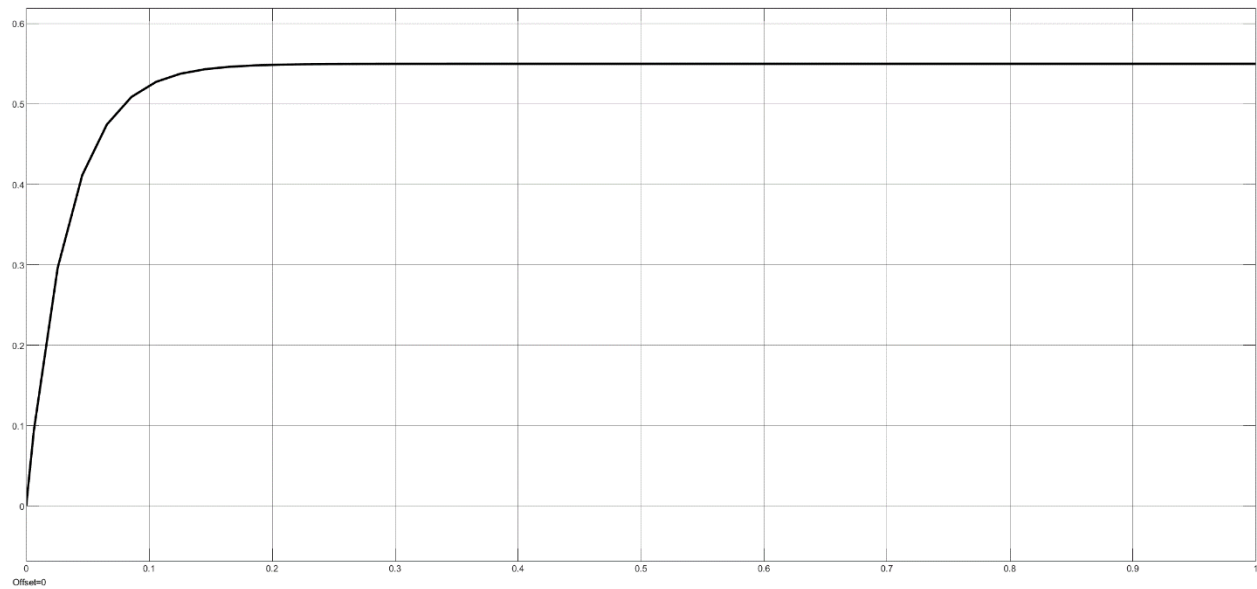
$$\tau(seconds) = 6 * 5.5ms = 33ms$$

در نتیجه تابع تبدیل سیستم مورد نظر برابر خواهد بود با:

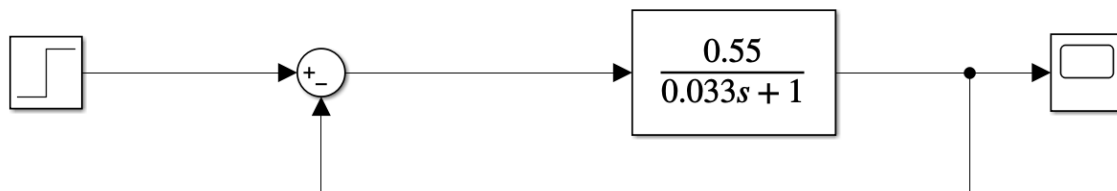
$$T(s) = \frac{K}{\tau s + 1} = \frac{0.55}{0.033s + 1}$$

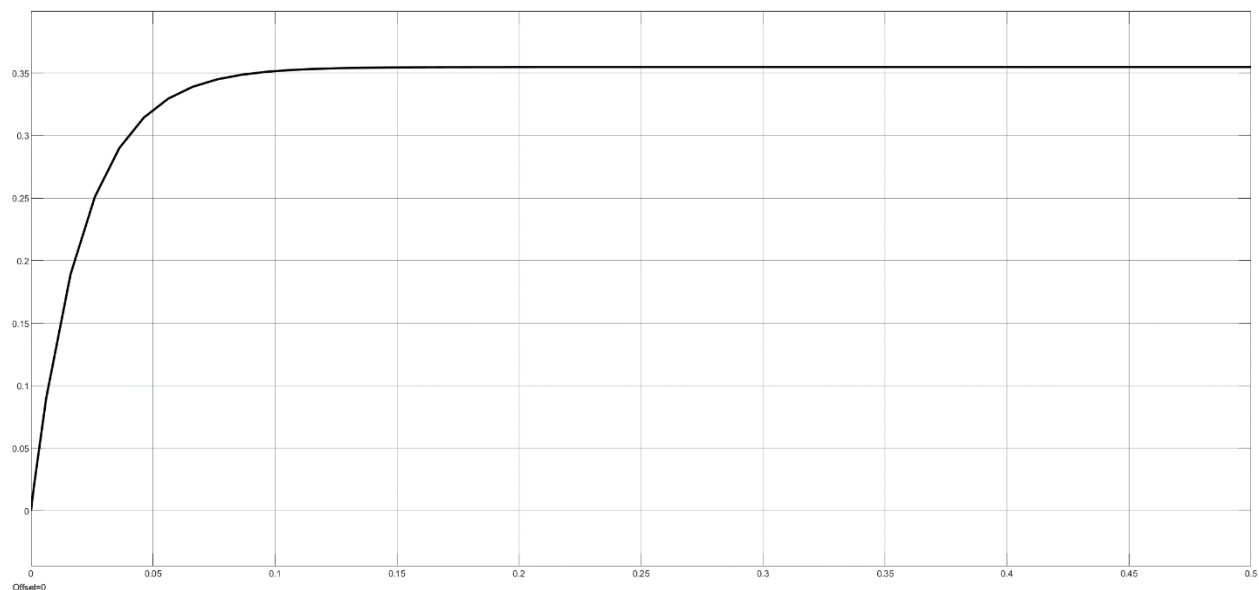
### سوال 3-9 کنترل حلقه بسته

حال با در اختیار داشتن تابع تبدیل سیستم، اقدام به طراحی یک کنترلر PID برای این سیستم می کنیم.

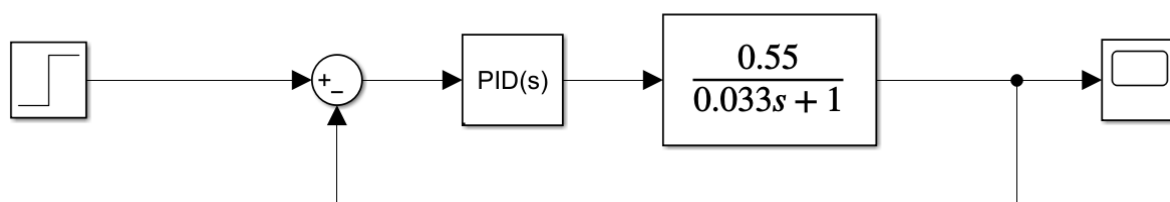


ابتدا پاسخ پله ی سیستم حلقه باز را در سیمولینک شبیه سازی می کنیم.  
سپس، پاسخ پله ی سیستم حلقه بسته را مطابق مدار زیر به دست می آوریم.





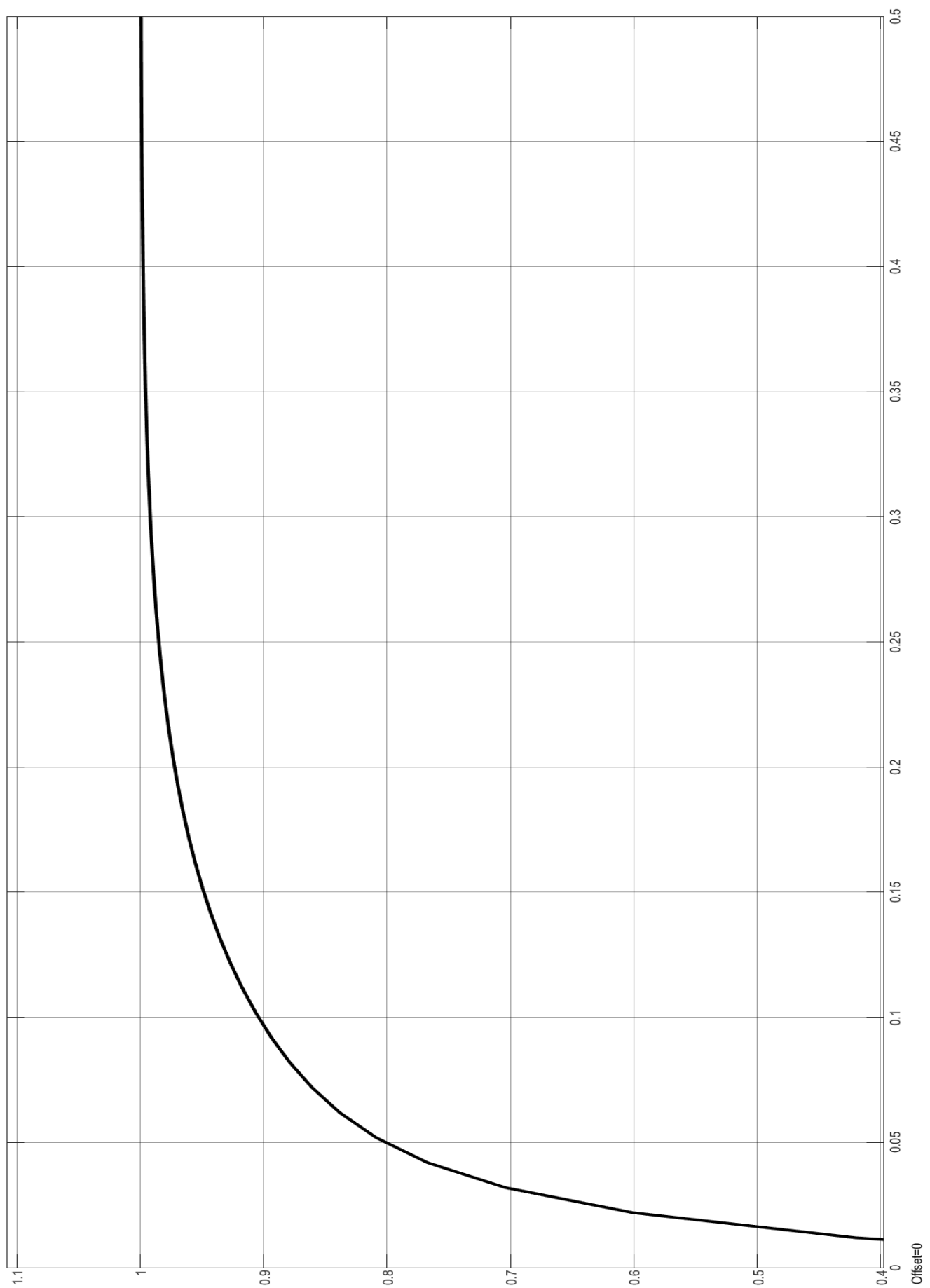
مشاهده می کنیم که این سیستم دارای خطای ماندگار زیادی است و نیاز داریم تا با یک کنترلر PID آن را کنترل کنیم.



ضرایب PID پس از آزمون و خطا برابر با مقادیر زیر است:

$$K_P = 3, K_I = 50, K_D = 0.0001$$

پاسخ پله ی سیستم کنترل شده به صورت زیر می باشد:





حال مقادیر به دست آمده را بر روی آردوینو پیاده می کنیم. برای این منظور در کد داده شده در دستور کار، مقادیر PID و Setpoint را جایگذاری می کنیم.

```
#include <PID_v1.h>
double Setpoint, Input, Output ,
error;
double Kp=3, Ki=50, Kd=0.0001; //
300,50,10
int pwm=10;
int sensor=5;
int duty;
const int sampleRate = 1; // Variable
that determines how fast our PID loop
runs
PID myPID(&Input, &Output, &Setpoint,
Kp, Ki, Kd, DIRECT);
void setup()
{
  Setpoint=110;
  duty=127;
  Serial.begin(9600);
  pinMode(pwm,OUTPUT);
  pinMode(sensor,INPUT);

  myPID.SetMode(AUTOMATIC); //Turn on
the PID loop
  myPID.SetSampleTime(sampleRate);
  //Sets the sample rate
}
void loop()
{
  Input=analogRead(A5);
  Input=map(Input,0,1023,0,255);
  error=Setpoint-Input;
  myPID.Compute(); //Run the PID loop
  duty=Output;
  analogWrite(pwm,duty);
  Serial.print(Input);
  Serial.print(" ");
  Serial.print(duty);
  Serial.print(" ");
  Serial.println(error);
}
```

در نمودار صفحه ی بعد، نتیجه ی آزمایش را می توان مشاهده کرد.

نمودار بالا که مربوط به Value1 است سیگنال ورودی، نمودار وسط که مربوط به Value2 است سیگنال فرمان یا همان duty cycle و سیگنال پایین مربوط به Value3 است مقدار خطا می باشد.

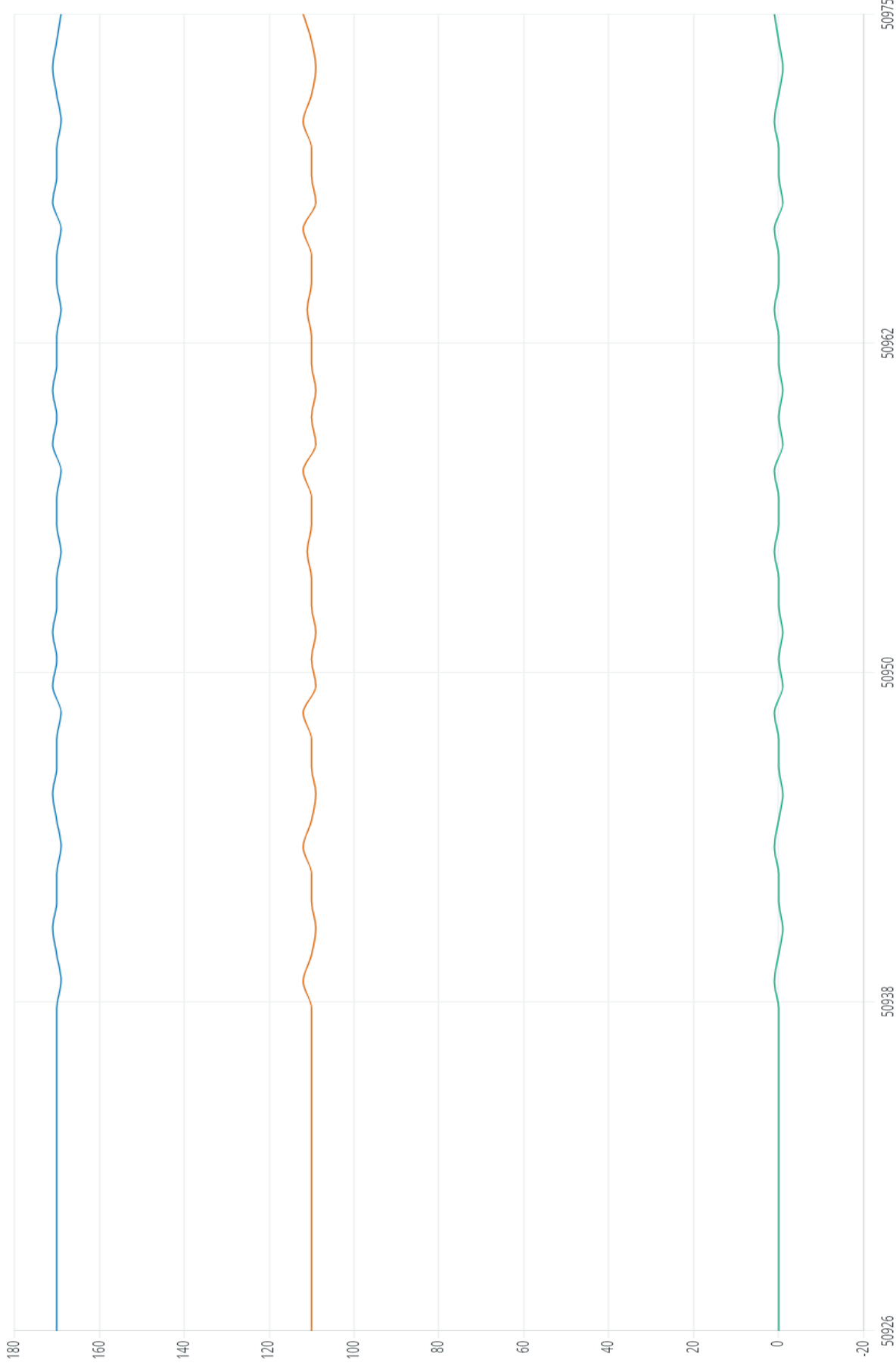
مطابق نمودار، مقادیر اندازه گیری شده به شرح زیر است:

Overshoot = 1.5%

Rise Time = 70ms

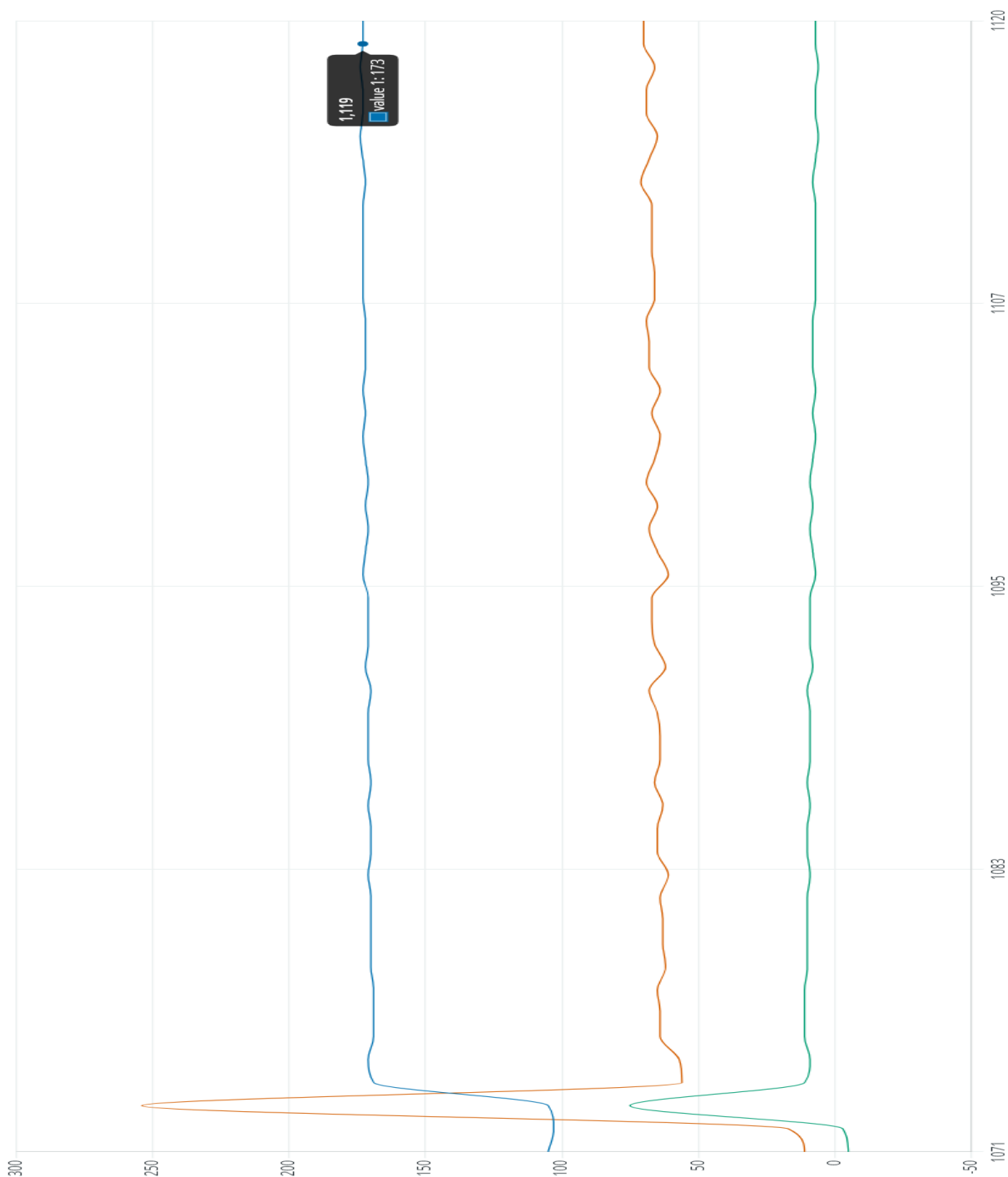
☒ value 1 ☒ value 2 ☒ value 3

Interpolate ☒ RUN



#### سوال 4-9: پاسخ حلقه بسته به ورودی

در این بخش از برنامه، هر یک ثانیه مقدار Setpoint را از 100 به 180 و برعکس تغییر می دهیم. نحوه ی تغییر خروجی به شکل زیر می باشد. همچنین ما اقدام به اندازه گیری مقدار فرافشار و سرعت پاسخ می کنیم.



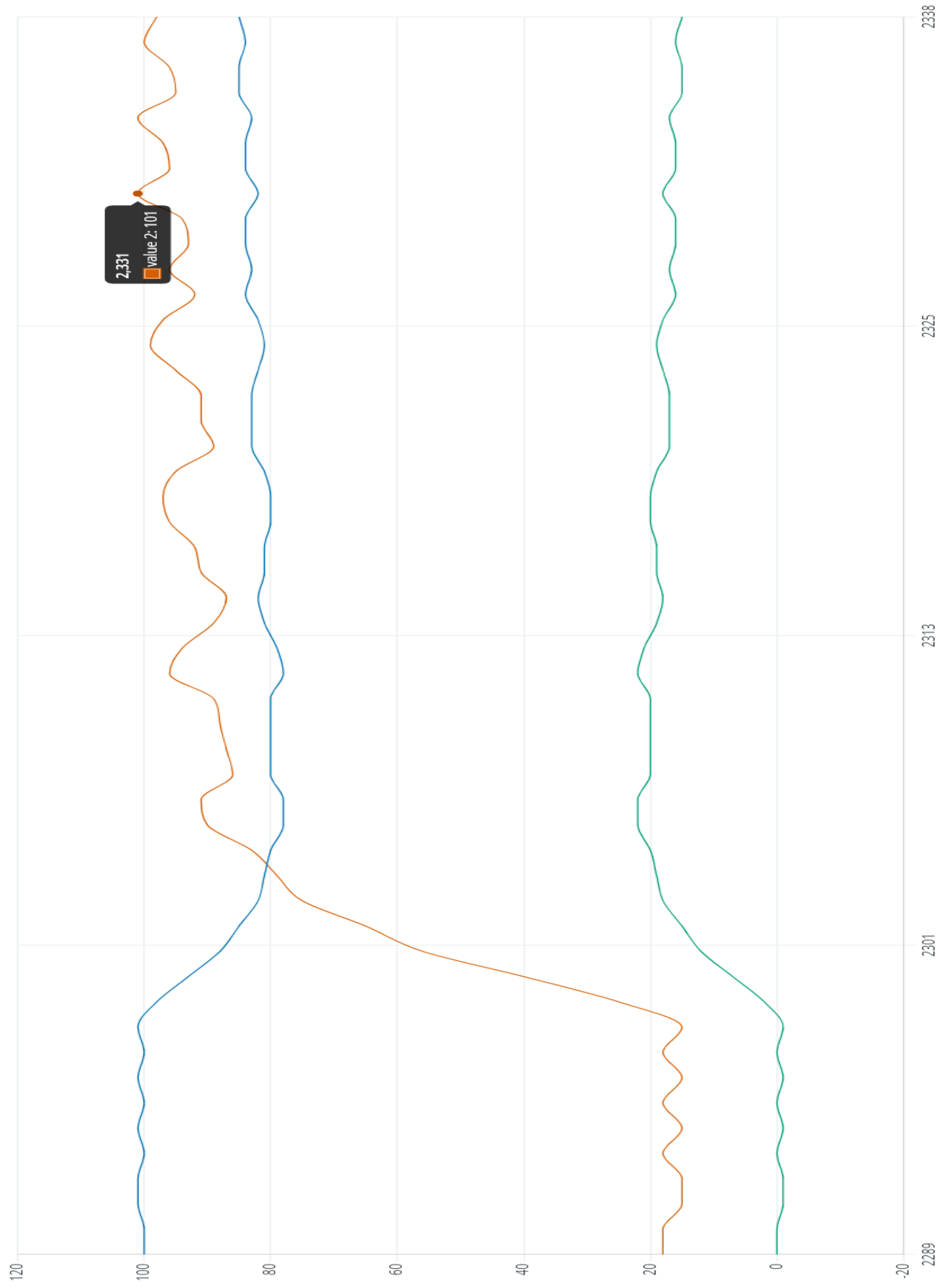
مطابق اندازه گیری ها مشاهده می شود که اورشوت سیستم برابر با  $1\% = \frac{1}{170-70}$  می باشد، اما مدت زمان آن برای رسیدن به مقدار نهایی تقریباً برابر با 240 سمپل که معادل با 4 ثانیه است می باشد. در نتیجه در طی یک ثانیه، به مقدار نهایی نرسیده و حدوداً  $3\% = \frac{6}{180}$  خطای ماندگار دارد.

#### سوال 5-9: پاسخ سیستم به اختلال:

در این قسمت ست پوینت را بر روی عدد 100 تنظیم کرده و با تغییر دادن محل LED، پاسخ سیستم را ثبت و ذخیره می کنیم.

به دلیل کند بودن سیستم، امکان تهیه ی عکس از تمام نمودار تا رسیدن به مقدار نهایی وجود ندارد. نمودار فوق، رفتار سیستم پس از اعمال اختلال را نشان می دهد.

همچنین، مطابق اندازه گیری های انجام شده، سیستم پس از حدود 4 ثانیه به Setpoint باز می گردد



ممنون از توجه حضرت عالی