Digital Control

LABORATORY REPORT

Position Control and PID controller

Amirkabir University of Technology
(Tehran Polytechnic)

Amirhossein Sheikholeslami
Mahboobeh Shakeri
Ahmadreza Nazari
Niloofar Raeesolsadati
Amin Razzaghi

Contents

Introduction

In this experiment, we try to implement our code in Web world with tools like flask , ajax , json and our final target is reaching to show sensor's data live .
First we try to make a web page, but the point is we use one of the web framework of python named Flask .
Then we try to send data and receive essential data or we can say receive proper response with processing and reading the data , but we want to all these works without refresh the web page so we use Ajax to do that . because Ajax has that ability to do .
Then we use json and process it . and the finally our final goal is showing the data of sensors live , without refreshing the webpage.

Encoder

the initial concept is that we count pulses from the encoder with interrupt and when we have the number of pulses of encoder within the whole length of our path(means from zero position to the max position) ,we can read the position of robot in any x-position according to the number of all pulses . in this task we use encoder port that exist on the device as input-port of Arduino .
but the point is that we finally noticed that we can't get position of robot by encoder because the frequency of the encoder less than the frequency of the Arduino and ... so we can't do the task with this encoder.

Control via output port

Read output voltage and map it

As it was said in Introduction section , instead of port of encoder , we use the output port of our robot , that it's voltage is proportional to the position of the robot .

You can see the setup part of Arduino code in figure 1 .

```
const int MVPin = A0;
const int outPin = 9;
int DV=500;
int minError = DV - 2700;
int maxError = DV;
int MV;
int MVD;
int error, errorLast;
float D = 0, I = 0;
float Ts = 0.01;
int u;
int Kp = 15, Kd = 0.1, Ki = 1;
void setup() {
  // put your setup code here, to run once:
  pinMode(MVPin, INPUT);
  pinMode(outPin, OUTPUT);
  Serial.begin(9600);
}
```

Figure 1: setup part of arduino code

In the setup part of code we don't have anything new , and we know it very well and doesn't need to explain .

```
void loop() {
  // put your main code here, to run repeatedly:
  MV = analogRead(MVPin);
  MVD = map(MV,0,1023,0,2700);
  errorLast = error;
  error = (DV - MVD);
  D = (error - errorLast)/Ts;
  I +=  error*Ts;
  if( I > 65535){
    I = 65535;
  }
  else if( I < -65535){
    I = -65535;
  }
  if(I * error < 0 ) {
    I = 0;
  }

  u = map((error*Kp) + (D*Kd) + (I*Ki), minError, maxError, 0, 255);
  if(u > 255)
    u = 255;
  else if(u < 0)
    u = 0;
  analogWrite(outPin, u);
  delay(Ts * 1000);
  char str[50];
  sprintf(str,"error: %d\tu: %d",error, u);
  Serial.println(str);
}
```

Figure 2: loop part of arduino code

In first step we read the input voltage of output port of robot . and then map it from zero to 1023 (because the analog input port of Arduino is 10bits) to zero till 2700 (because the full length of robot is 27 cm) .

The for controlling the robot we must declare a set point , that we swt it to 5 cm and we define our error with (setpoint – current$_p osition$).

In the experiment we start the control of robot with kp and then we add one integraler and then we append a derivator controller to it , but for summery we Just explain PID control and the last part of our design .

For making Derivative we must subtract the last error from the current error and then devide into the sample priode that we defined above of our code .

And furthermore for making integrator we must multiply error to sample period
But the point of integrator is we must consider wind-up feature of it . for solving this problem we must add some if , that if amount of integrator will more or less than the our limit number then set it to the numbers that we say .
And also for solving the reset the integrator (means if the sign of error will change for example from positive to negative) , the integrator number will set to zero .

And in final step we sum integrator and derivative and gain with together and then send it as final error signal to robot for compensating and reaching to the set point that we defined .