제어시스템 모델링

[실습1-2 Bump test]

이름 : 김 용 현

학번 : 2017006262

|  |
| --- |
| 실습이론 |
| DC 모터  전달함수 G(s) = S \* Seta\_m(s) / V\_a(s) = K\_t / (R\_a \* J\_s + K\_t \* K\_e)  저역통과필터(Low Pass Filter)  특정 주파수 이상의 신호를 감쇠 시켜 차단 주파수 이하의 신호만 통과시키는 필터이다. 본 실습에서는 모터의 속도에 생긴 잡음을 제거하기 위하여 사용된다. 입력 신호=X(t), 출력 신호=Y(t), 차단 주파수=Wc 일 때  저역통과필터를 적용하여 나온 그래프를 이용하면 전달함수의 K값과 tau값을 추정할 수 있다.  G(s) = K / (tau \* S +1)  #K값과 tau값을 매트랩에 직접 입력해가며 근사값 추정 |

|  |
| --- |
| 실습코드(아두이노) |
| #include <SSD1306.h>  #include <MsTimer2.h>  // OLED Setup  #define OLED\_DC 5  #define OLED\_CLK 8  #define OLED\_MOSI 7  #define OLED\_RESET 6  SSD1306 oled(OLED\_MOSI, OLED\_CLK, OLED\_DC, OLED\_RESET, 0);  //MOTOR driver pin  #define PWM 9  #define IN1 10  #define IN2 11  //Encoder Pin  #define ENCODER\_A 3  #define ENCODER\_B 2  float t = 0;  unsigned int time\_val = 0;  float Ts = 0.001;  float End\_Time = 1.0;  void Set\_PWM(int motor)  {  if (motor >= 0)  {  digitalWrite(IN1, LOW);  digitalWrite(IN2, HIGH);  analogWrite(PWM, motor);  }  else  {  digitalWrite(IN1, HIGH);  digitalWrite(IN2, LOW);  analogWrite(PWM, motor);  }  }  int cart\_encoder = 0;  float cart\_position = 0.0;  float cart\_velocity = 0;  float filtered\_velocity = 0;  float prev\_cart\_position = 0.0;  float prev\_filtered\_velocity = 0.0;  void control() {  cart\_position = float(cart\_encoder) / 520 \* PI;  float cart\_velocity = (cart\_position - prev\_cart\_position) / Ts;  float filtered\_velocity = ((0.97560975609756097560975609756098) \* prev\_filtered\_velocity) + ((0.02439024390243902439024390243902) \* cart\_velocity);  prev\_cart\_position = cart\_position;  prev\_filtered\_velocity = filtered\_velocity;  if (t > End\_Time)  Set\_PWM(0);  else  Set\_PWM(255);  t += Ts;  // Serial Write  byte \* time\_ = (byte \*) &t;  byte \* filtered\_velocity\_ = (byte \*) &filtered\_velocity;  Serial.write(time\_, 4);  Serial.write(filtered\_velocity\_, 4);  }  void setup() {  //display setup  int fff = 1;  TCCR1B = (TCCR1B & 0xF8) | fff;  oled.ssd1306\_init(SSD1306\_SWITCHCAPVCC);  oled.clear(); // clears the screen and buffer  //motor driver setup  pinMode(IN1, OUTPUT);  pinMode(IN2, OUTPUT);  pinMode(PWM, OUTPUT);  //Encoder Setup  pinMode(ENCODER\_A, INPUT);  pinMode(ENCODER\_B, INPUT);  Serial.begin(2000000);  delay(200);  //Timer Setup  MsTimer2::set(1, control);  MsTimer2::start();  //Interrupt Setup  attachInterrupt(0, doEncoderA, CHANGE);  attachInterrupt(1, doEncoderB, CHANGE);  // motor STOP  digitalWrite(IN1, 0);  digitalWrite(IN2, 0);  digitalWrite(PWM, 0);  oled.drawstring(00, 2, "BUMP\_TEST");  oled.display();  }  void loop() {  }  void doEncoderA() {  if(digitalRead(ENCODER\_A) == digitalRead(ENCODER\_B))  {  cart\_encoder = cart\_encoder +1;  }  else  {  cart\_encoder = cart\_encoder -1;  }  }  void doEncoderB() {  if(digitalRead(ENCODER\_A) == digitalRead(ENCODER\_B))  {  cart\_encoder = cart\_encoder -1;  }  else  {  cart\_encoder = cart\_encoder +1;  }  } |
| 실습코드(matlab) |
| close all;  clear;  %% Data Acquisition  % Serial Port Setup  port\_name = "COM1";  baud\_rate = 2000000;  s = serialport(port\_name,baud\_rate);  % Parameter Setup  endTime = 1;  Ts = 0.001;  % Data list  time\_list = zeros(endTime/Ts,1);  vel\_list = zeros(endTime/Ts,1);  for i = 1:1:size(time\_list)  time = read(s,1,"single");  time\_list(i) = time;  vel = read(s,1,"single");  vel\_list(i) = vel;  end  % 연결종료  s.delete  %% DRAW STEP RESPONSE  s = tf("s");  K = 43; % 실제 값과 유사한 값을 찾으시오  tau = 0.09; % 실제 값과 유사한 값을 찾으시오  G = K/(tau\*s+1); % 1차 표준형 시스템  % 그리기  figure(1)  plot(time\_list,vel\_list,'r-');  hold on;  step(G) |

|  |
| --- |
| 결론 |
| Bump test 실행 결과 그래프에 빨간 선으로 결과가 도출되었으며 Matlab을 이용하여 K값과 tau값을 추정하여 K=43, Tau=0.09를 대입하였을 때 비슷한 형태의 그래프가 그려지는 것을 보아 K=43, tau=0.09로 값을 도출해 낼 수 있었다.  실습 1-1에서 작성했던 pwm, encoder 코드를 활용하고 전달함수를 이용한 저역통과필터 함수를 추가하여 실습을 진행하였다. 1-2 진행 시 1-1에서 encoder 함수를 적용하지 않아 그래프 값이 0으로 고정되는 오류가 있었다.  저역통과필터의 설계를 통해 특정 주파수 이상의 값은 차단하고 낮은 주파수의 값을 흐르도록 하는 방법에 대해 배우게 되었다. |