제어시스템 모델링

[실습1-3 Bode Plot]

이름 : 김 용 현

학번 : 2017006262

|  |
| --- |
| 실습이론 |
| Bode Plot  시스템 입력 주파수를 증가시켜 출력 주파수의 크기 비율을 Logscale로 그린 그래프를 bode plot이다.  실습에서 주파수를 바꾸면서 그래프를 그리지 않고 chirp signal 사용.  \*chirp signal: 주파수가 시간에 따라 선형적으로 변하는 신호.  주파수 영역 분석 |

|  |
| --- |
| 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;  float u = 0;  float Sampling\_Time = 0.001;  float Start\_Frequency = 0;  float End\_Frequency = 200;  float End\_Time = 20.0;  int cart\_encoder = 0;  float cart\_position = 0.0;  float prev\_cart\_position = 0.0;  float prev\_filtered\_velocity = 0.0;  float cart\_velocity = 0;  float filtered\_velocity = 0;  void Set\_PWM(int motor)  {  if (motor >= 0)  {  digitalWrite(IN1, HIGH);  digitalWrite(IN2, LOW);  analogWrite(PWM, motor);  }  else  {  digitalWrite(IN1, LOW);  digitalWrite(IN2, HIGH);  analogWrite(PWM, motor);  }  // Fill in the Missing Code  }  float getChirpSignal(float Start\_Frequency, float End\_Frequency, float Sampling\_Time, float End\_Time , float t) {  // Fill in the Missing Code, HINT: f(t) = ct+f0 , c = f1-f0/End\_Time , x(t) = sin(2pi(c/2\*t^2+f0\*t)  //  float value = 0;  float c = (End\_Frequency - Start\_Frequency) / (End\_Time);  return sin(2 \* 3.141592 \* (c/2\*t\*t));  }  void control() {  cart\_position = float(cart\_encoder) / 520 \* PI;  float cart\_velocity = (cart\_position - prev\_cart\_position) / Sampling\_Time;  filtered\_velocity = cart\_velocity;  prev\_cart\_position = cart\_position;  // Fill in the Missing Code  //  u = getChirpSignal(Start\_Frequency, End\_Frequency, Sampling\_Time , End\_Time, t);  if (t > End\_Time)  Set\_PWM(0);  else  Set\_PWM(u \* 255);  t += Sampling\_Time;  // 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, "BODE\_PLOT");  oled.display();  }  void loop() {  }  void doEncoderA() { // Fill in the Missing Code  if(digitalRead(ENCODER\_A) == digitalRead(ENCODER\_B))  {  cart\_encoder = cart\_encoder +1;  }  else  {  cart\_encoder = cart\_encoder -1;  }    }  void doEncoderB() { // Fill in the Missing Code  if(digitalRead(ENCODER\_A) == digitalRead(ENCODER\_B))  {  cart\_encoder = cart\_encoder -1;  }  else  {  cart\_encoder = cart\_encoder +1;  }    } |

|  |
| --- |
| 실습코드(matlab) |
| clear  close all;  %% Data Acquisition  % Serial Port Setup  port\_name = "COM1";  baud\_rate = 2000000;  s = serialport(port\_name,baud\_rate);  % Parameter Setup  Ts = 0.001;  End\_Time = 20;  Start\_Frequency = 0;  End\_Frequency = 200;  % Data list  time\_list = zeros(End\_Time/Ts,1);  vel\_list = zeros(End\_Time/Ts,1);  filtered\_vel=0;  for i = 1:1:size(time\_list)  time = read(s,1,"single");  vel = read(s,1,"single");  vel\_list(i) = vel;  time\_list(i) = time;  end  s.delete  Ts = 0.001;  t = 0:Ts:(End\_Time-Ts);  x =chirp(t,Start\_Frequency,End\_Time,End\_Frequency)';  y = vel\_list;  L = length(y);  f = linspace(0,(1/Ts)\*pi,L/2+1);  fft\_x = fft(x)  P2\_in = abs(fft\_x /L);  P1\_in = P2\_in(1:L/2+1);  P1\_in(2:end-1) = 2\*P1\_in(2:end-1);  %Fast Fourier Transform for output signal  fft\_y= fft(y);  P2\_out = abs(fft(y)/L);  P1\_out = P2\_out(1:L/2+1);  P1\_out(2:end-1) = 2\*P1\_out(2:end-1);  % draw bodeplot  semilogx(f,20\*log10((P1\_out./P1\_in)),"r")  hold on;  % Find Parameter  s = tf("s");  K =30;  tau =0.05;  G = K/(tau\*s+1);  bode(G); |
| 결론 |
| Chirp신호 입력, 출력속도에 대한 Bode plot을 출력하였고 이에 대한 K와 tau를 추정한 결과 K=30, tau=0.05가 나오게 되었다.  실습 1-2의 아두이노 코드에서 pwm, control함수를 bode plot에 맞게 수정하는 것으로 실습을 진행할 수 있었다.  Frequency response 함수를 이용하여 bode plot 실습을 진행하였다. F(t), X(t) 함수를 아두이노 코드에 대입하여 input signal FFT의 magnitude 값과 output signal FFT의 magnitude값을 이용하여 bode plot을 그릴 수 있었다. |

