Anexo 1: Pruebas en el dedo Pulgar

A continuación se presenta la respuesta al escalón del sistema, se observa que se asemeja a un sistema de segundo orden. Para hallar los parámetros de referencia PID del sistema se utilizó la herramienta Matlab para luego realizar pruebas sobre la planta con el fin de obtener los parámetros más apropiados. En la figura 1, se presenta la respuesta al escalon de la planta, la cual se simulará en Simulink con el fin de obtener mediante la herramienta de ayuda de sintonización del Matlab los parámetros.

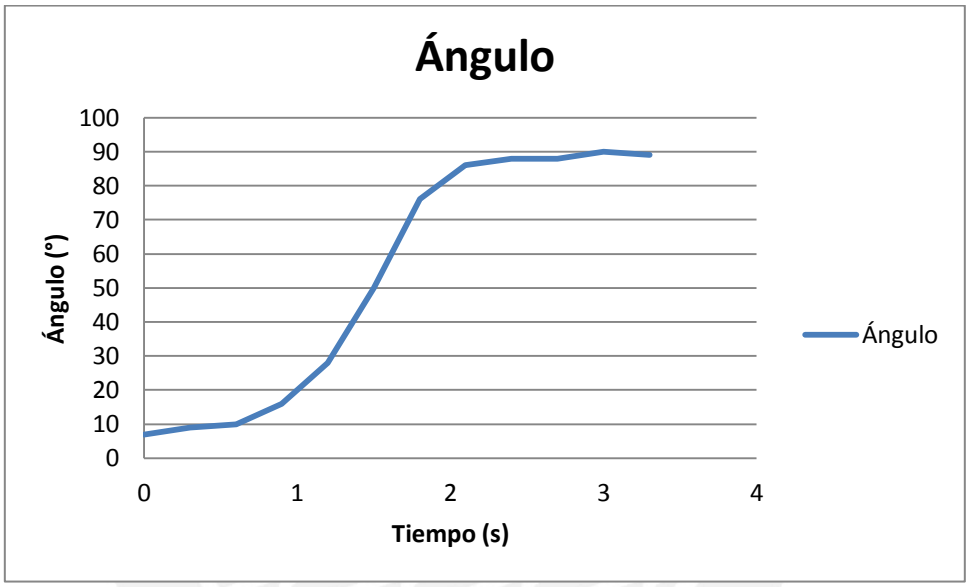

Figura 1: Respuesta al escalón de la planta. Fuente: Elaboración propia

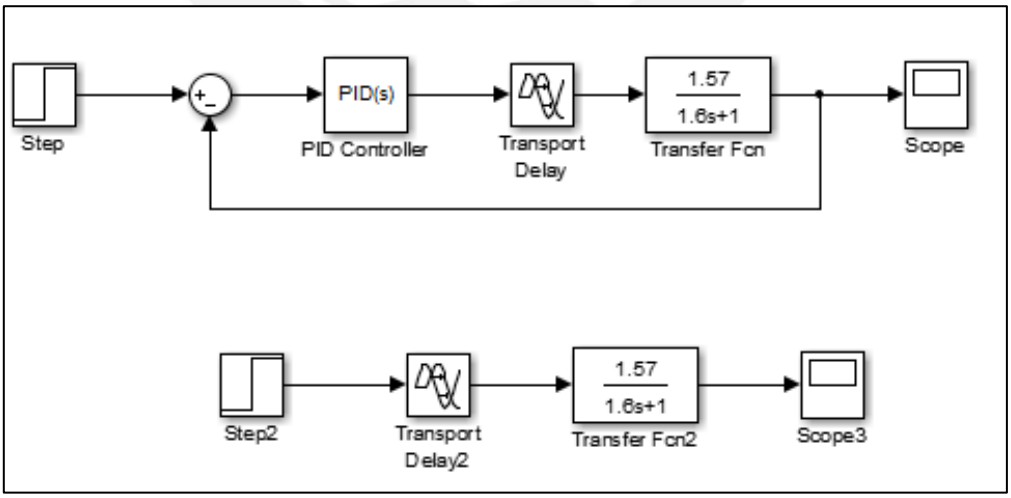

Figura 2: Simulación para la obtención de los parámetros PID. Fuente: Elaboración propia

Los parámetros de la simulación son los siguientes:

Kp= 0.9

Ki=0.54

Kd=0.13

Para discretizar los parámetros debemos elegir el tiempo de muestreo aplicando el criterio mencionado en [45]:

$$\frac{1}{15} * T_{95\%} < T < \frac{1}{4} * T_{95\%}$$

Dónde:

$T_{95\%}$: Tiempo que demora a llegar al 95% de su valor final

T: Periodo de muestreo

Según la gráfica, en la figura ,3 podemos observar que el tiempo que demora en llegar al 95% de su valor final es de 1.4 segundos. Aplicando la formula el tiempo de muestreo debe estar entre 0.35 y 0.093 segundos. Como en el caso del dedo índice se tomará como tiempo de muestreo 300 ms.

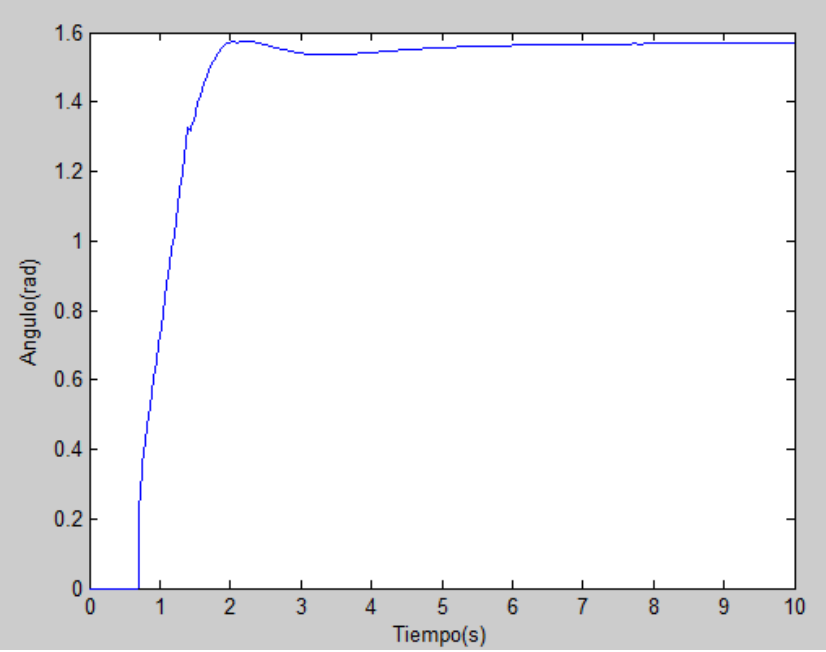

Figura 3: Respuesta del sistema real sintonizado respecto al tiempo. Fuente: Elaboración propia

Entonces los valores de los parámetros sintonizados serán:

Kp= 0.9

Ki=0.16

Kd=0.43

Realizando pruebas sobre la planta para afinar estos parámetros y obtener un tiempo de subida aproximadamente de 1.3 segundos, similar al tiempo que demora un dedo en realizar estos movimientos, se obtuvieron los siguientes:

Kp= 0.87

Ki=0.15

Kd=0.39

Se realizaron pruebas a diferentes ángulos y con otros pesos a los mostrados en el documento principal con el fin de comprobar el funcionamiento del sistema y control.

En la siguiente figura se muestra la variación del ángulo final entre 20 y 22 grados, en la figura 5, se muestra los valores captados por el sensor al momento de la realización de la prueba.

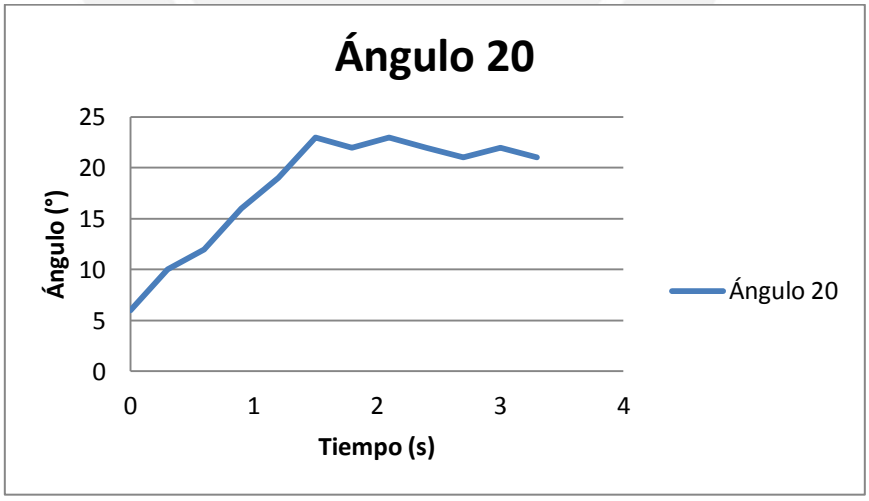

Figura 4: Kp=0.87, Ki=0.15, Kd=0.39. Fuente: Elaboración propia

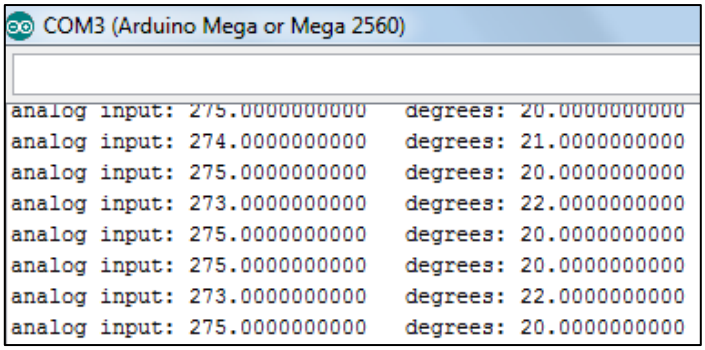

Figura 5: Kp=0.87, Ki=0.15, Kd=0.39. Fuente: Elaboración propia

Pruebas con pesa de 200gr en el dedo pulgar, en las gráficas podemos observar que el error en estado estable aumenta en 3 grados más en comparación de la prueba sin error.

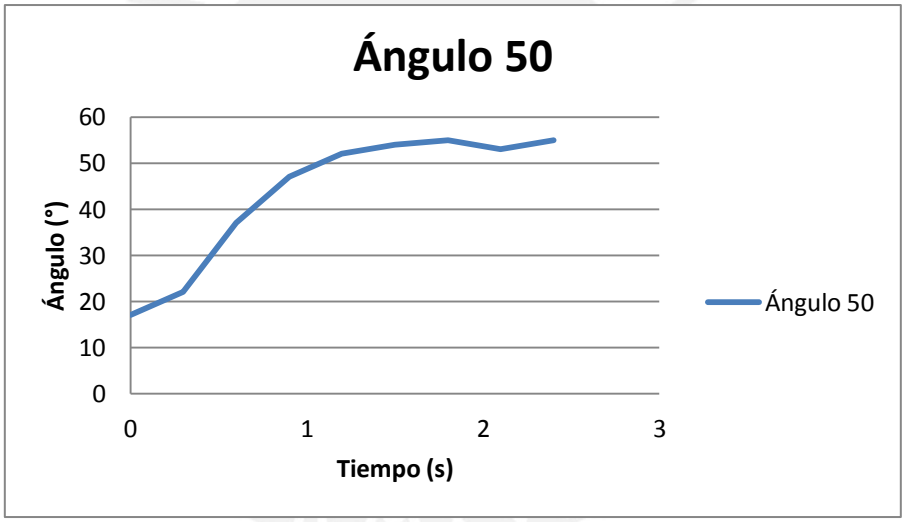

Figura 6: Kp=0.87, Ki=0.15, Kd=0.39. Fuente: Elaboración propia

Anexo 2: Programación

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

int lcd_key     = 0;

int adc_key_in  = 0;

int left=0; //decenas

int right=0; //unidades

int N=0;// variable de ángulo final

int cont=0;

#define btnRIGHT  0

#define btnUP     1

#define btnDOWN   2

#define btnLEFT   3

#define btnSELECT 4

#define btnNONE   5

int read_LCD_buttons(){               //Lectura de los botones

   adc_key_in = analogRead(0);       // Lectura del valor del sensor

   if (adc_key_in > 1000) return btnNONE;

    if (adc_key_in < 50)   return btnRIGHT;

    if (adc_key_in < 195)  return btnUP;

    if (adc_key_in < 380)  return btnDOWN;

    if (adc_key_in < 555)  return btnLEFT;

    if (adc_key_in < 790)  return btnSELECT;
```

```
    return btnNONE;              // si no se presiona ningun boton retorna btnNONE
}
void setup(){
Serial.begin(9600);   //inicializamos la comunicacion serial con el motivo de visualizar los valores leidos por el sensor
int freq=1000;        //inicializamos el Duty cycle a una frecuencia de 1000 Hz
int duty;
long top;
long aux;
long N=256;
long Fclock=16000000;
pinMode(2, OUTPUT);
pinMode(3, OUTPUT);
pinMode(11, OUTPUT);
pinMode(12, OUTPUT);
pinMode(22,OUTPUT);
aux=Fclock/(freq*N) -1;
ICR3=aux;
ICR1=aux;
TCCR3A=0;
TCCR3B=0;
TCCR3C=0;
```

```
TCCR1A=0;

TCCR1B=0;

TCCR1C=0;

    TCCR3A = _BV(COM3A1)  |_BV(COM3B1)|_BV(COM3C1) | _BV(WGM31);

    TCCR3B = _BV(WGM33) | _BV(WGM32) | _BV(CS32);

    TCCR1A = _BV(COM1A1)  |_BV(COM1B1)|_BV(COM1C1) | _BV(WGM11);

    TCCR1B = _BV(WGM13) | _BV(WGM12) | _BV(CS12);

     lcd.begin(16, 2);               // inicializamos la libreria

   lcd.setCursor(0,0);             // configura la posición del corsor en el LCD

   lcd.print("Bienvenidos");      // Impresión del mensaje en el LCD

     lcd.setCursor(0,1);

   lcd.print("Press select");

}

void loop() {

lcd_key = read_LCD_buttons();

lcd_key = read_LCD_buttons();

  switch(lcd_key)

 {

   case btnSELECT:

   {

     lcd.clear();

     lcd.setCursor(0,0);

     lcd.print("ING ANGULO");
```

```
      lcd.setCursor(0,1);
      lcd.print("ING POS FIN");
       eleccion();
       break;
       }
 }
}
void eleccion(){
double ei=0;// variables del PID
double eant=0;// variables del PID
double dut;// variables del PID
double er=0;// variables del PID
double Ki;// variables del PID
double Kp;// variables del PID
double Kd;// variables del PID
double duty;// variables del PID
double e;// variables del PID
double ed;// variables del PID
double ef;// variables del PID
double sensor1;
lcd.setCursor(0,0);
 lcd_key = read_LCD_buttons();
  while(true)
```

```
{
  lcd_key = read_LCD_buttons();
  switch(lcd_key)
   {
    case btnUP:
    {
      lcd.clear();
      lcd.setCursor(0,0);
      lcd.print("INGRESA ANG");
      goto numero;
      break;
        }
    case btnDOWN:
      {
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("POS DEF");
        goto alternativas;
        break;
         }
   }
}
numero:
```

```
while(true)
 {
  lcd_key = read_LCD_buttons();
  switch(lcd_key)
  {
  case btnLEFT:
   {
     left=left+1;
     delay(300);
     break;
      }
  case btnRIGHT:
     {
       right=right+1;
       delay(300);
       break;
        }
   case  btnUP:
   {
   lcd.clear();
   lcd.setCursor(0,0);
   lcd.print("realizando");
   N=left*10+right;
```

```
    goto sintonizacion2;    //Manda la variable final leida del LCD a la funcion sintonización 2 con el fin de empezar con el PID del dedo pulgar
   break;
    }
    case btnSELECT:
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("realizando");
    N=left*10+right;
    goto sintonizacion;    //Manda la variable final leida del LCD a la funcion sintonización con el fin de empezar con el PID del dedo indice
   break;
   }
    N=left*10+right;
  lcd.setCursor(0,1);
  lcd.print(N);
   }
alternativas:
delay(300);
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("SUBIR INDICE");
  lcd.setCursor(0,1);
```

```
 lcd.print("SUBIR PULGAR");
while(true)
 {
 lcd_key = read_LCD_buttons();
   switch(lcd_key)
 {
   case btnUP:
 { lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("SUB INDICE");
 N=45;
goto sintonizacion;
break;
   }
   case btnRIGHT:
   {
 lcd.clear();
 lcd.setCursor(0,0);
 lcd.print("MOVI. AVANZ 1");
 lcd.setCursor(0,1);
 lcd.print("MOVI. AVANZ 2");
 delay(200);
    while(true)
```

```
{
 lcd_key = read_LCD_buttons();
 switch(lcd_key)
 {
   case btnUP:
   {

    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("MOVI. AVANZ 1");
    N=80;
    goto sintonizacion;
  break;
    }
      case btnDOWN:
  {
       lcd.clear();
     lcd.setCursor(0,0);
     lcd.print("MOVI. AVANZ 2");
     N=80;
     goto sintonizacion2;
  break;
       }
```

```
        }
       }
     }
      case btnDOWN:
     {
      lcd.clear();
     lcd.setCursor(0,0);
     lcd.print("SUBIR PULGAR");
     N=45;
     goto sintonizacion2;
      break;
      }
      }
     }
    sintonizacion2:
     while(true)
     {
     double sensor, degrees;
      sensor = analogRead(3);
      degrees = map(sensor, 296, 204, 0, 90);
      Serial.print("analog input: ");
      Serial.print(sensor,DEC);
      Serial.print("   degrees: ");
```

```
 Serial.println(degrees,DEC);

Kd=0.39;

Kp=0.87;

Ki=0.15;

ef=(3.1416/180)*N; //angulos en radianes

sensor1=(296-sensor)/58.5;

e=ef-sensor1; //la diferencia entre lo leido del lcd y el sensor

ed=e-eant;

ei=e+ei;

dut=Kp*e+Ki*(ei)+Kd*(ed);

eant=e;

 if(dut >= 1)

 {

   if(N >= 65)

   {dut=0.95;

    goto next;  }

   dut=0.7;

   }

 if(dut<=0)

 {

   dut=0.2;

   }

 next:
```

```
duty=dut*100;

lcd.clear();

lcd.setCursor(0,1);

lcd.print(int(duty));

OCR1B=ICR1*int(duty)/100;

delay(300);

 OCR1B=ICR1*int(duty)/100;    //utilizamos el timer 1, pin 12


 if(cont == 99) //Despues de 30 segundos se activa el ventilador

 {

   while(true)

   {

  digitalWrite(12,LOW);

  digitalWrite(22,HIGH);//recordar que el pin 2, es el que tiene la programacionpara la
activacion del ventilador

  double sensor, degrees;

  sensor = analogRead(2);

  degrees = map(sensor, 296, 204, 0, 90);

  Serial.print("analog input: ");

  Serial.print(sensor,DEC);

  Serial.print("   degrees: ");

  Serial.println(degrees,DEC);

  delay(300);
```

```
  }
 }
 cont=cont+1;
}
sintonizacion :
 while(true)
 {
 double sensor, degrees;
  sensor = analogRead(2);
  degrees = map(sensor, 240, 204, 0, 90);
  Serial.print("analog input: ");
  Serial.print(sensor,DEC);
  Serial.print("   degrees: ");
  Serial.println(degrees,DEC);
Kd=0.28;
Kp=0.7;
Ki=0.182;
ef=(3.1416/180)*N; //ángulo en radianes
sensor1=(240-sensor)/23;
e=ef-sensor1;
ed=e-eant;
ei=e+ei;
dut=Kp*e+Ki*(ei)+Kd*(ed);
```

```
eant=e;
 if(dut >= 1)
 {
   if(N >= 65)
   {dut=0.95;
    goto siguiente;  }
   dut=0.7;
   }
 if(dut<=0)
 {
   dut=0.2;
   }
 siguiente:
duty=dut*100;
lcd.clear();
lcd.setCursor(0,1);
lcd.print(int(duty));
OCR3B=ICR3*int(duty)/100;
delay(300);
OCR3B=ICR3*int(duty)/100;    //utilizamos el timer 3, pin 2
 if(cont == 99)    //Despues de 30 segundos se activa el ventilador
 {
   while(true)
```

```
  {
  digitalWrite(2,LOW);
  digitalWrite(22,HIGH);//recordar que el pin 2, es el que tiene la programacionpara la
activacion del ventilador
  double sensor, degrees;
  sensor = analogRead(2);
  degrees = map(sensor, 240, 204, 0, 90);
  Serial.print("analog input: ");
  Serial.print(sensor,DEC);
  Serial.print("   degrees: ");
  Serial.println(degrees,DEC);
  delay(300);
  }
  }
   cont=cont+1;
}
  exit:
return;
}
```

Anexo 3: Pruebas en el dedo índice

A continuación se presentará otras pruebas realizadas a diferentes ángulos en el dedo índice. Como se observa en las figuras el tiempo de subida es aproximadamente 1.3 segundos en todos los casos, el error en estado estable varía entre 2 a 3 grados según los el caso, pero este para efectos visuales en la posición final del dedo no es perceptible.

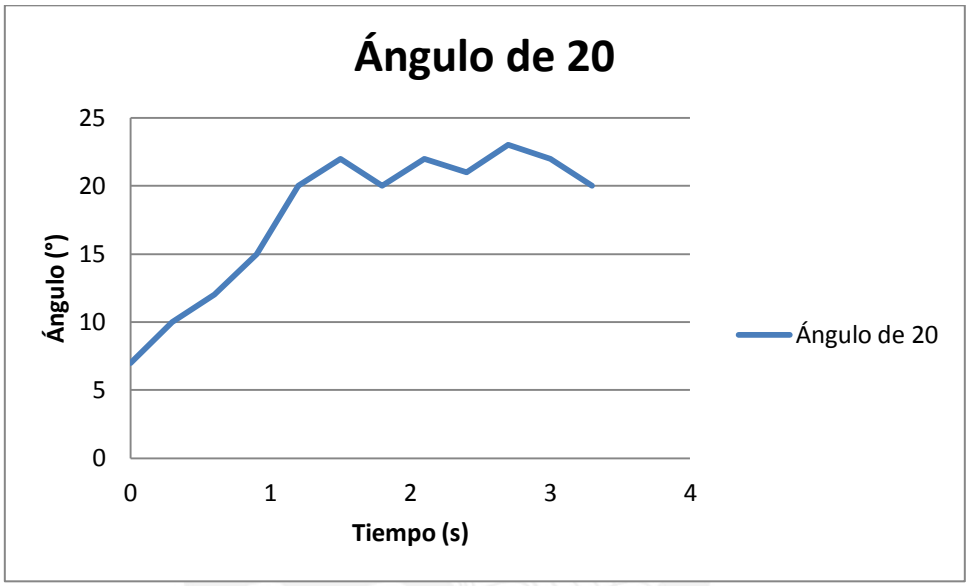

Figura 1: Kp=0.7, Ki=0.182, Kd=0.28. Fuente: Elaboración propia

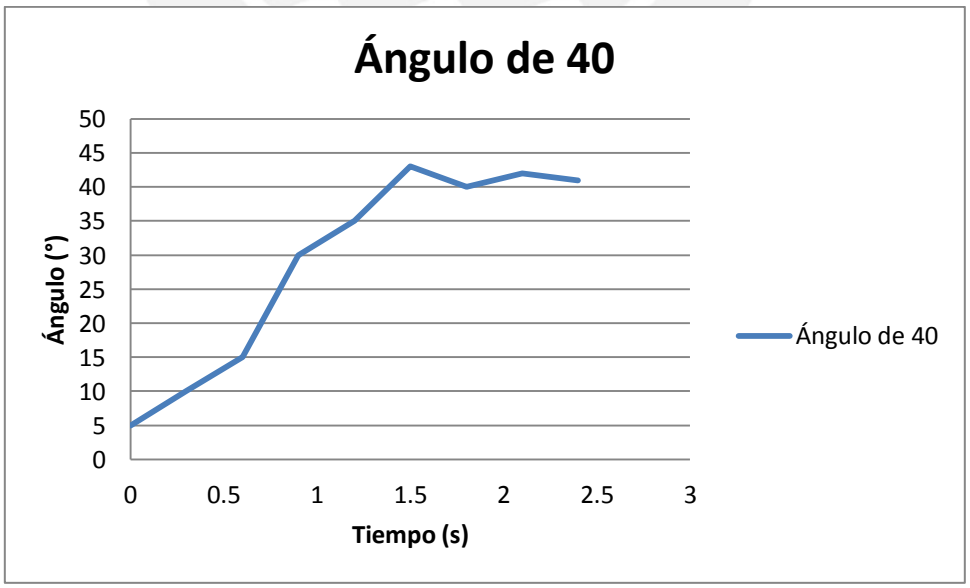

Figura 2: Kp=0.7, Ki=0.182, Kd=0.28. Fuente: Elaboración propia

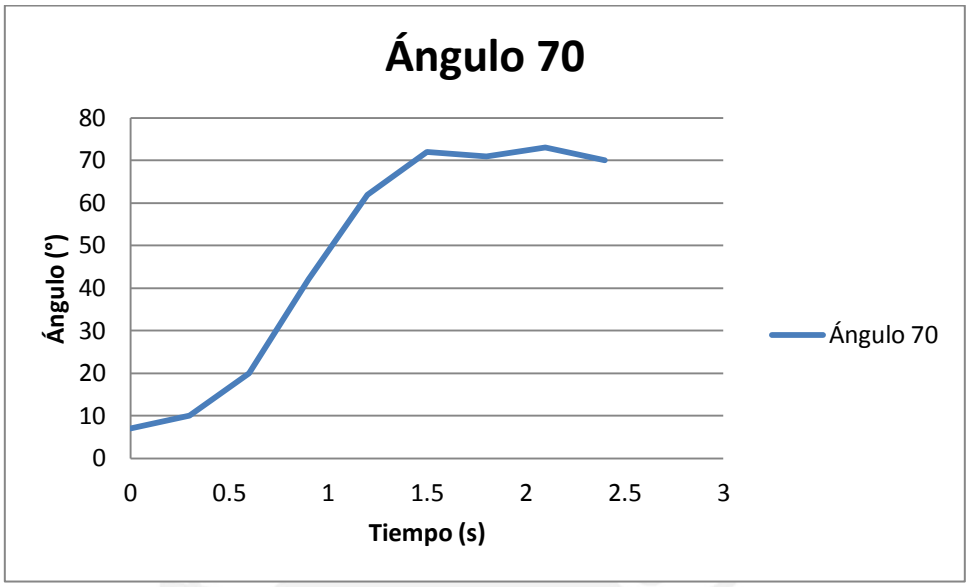

Figura 4: Kp=0.7, Ki=0.182, Kd=0.28. Fuente: Elaboración propia

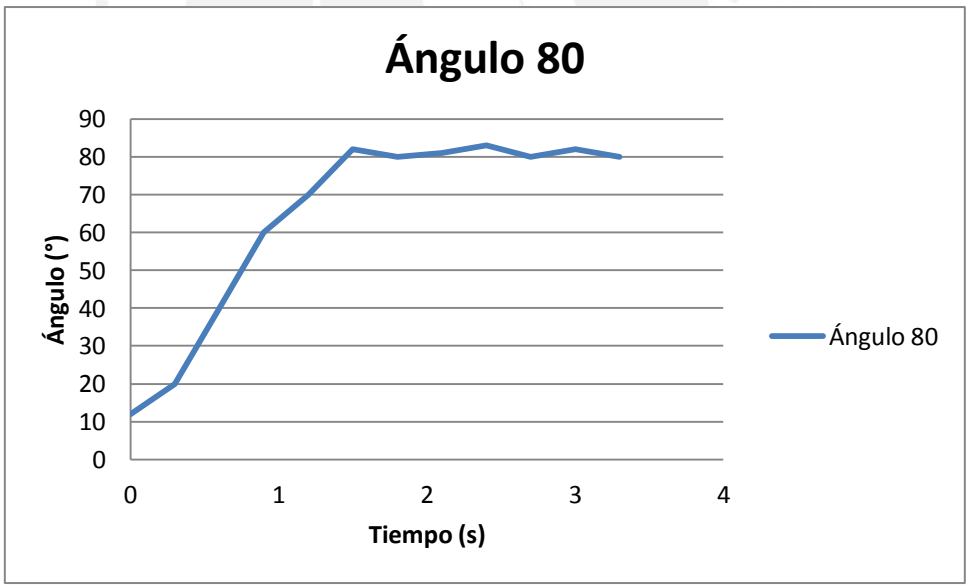

Figura 5: Kp=0.7, Ki=0.182, Kd=0.28. Fuente: Elaboración propia

Por otro lado, también se realizaron pruebas de 100 g en el dedo índice, donde se observa que el tiempo de subida aumenta en 0.3 segundos, en comparación a la prueba sin peso adicional; además, se tiene un aumento de 3 grados en el error en estado estable.

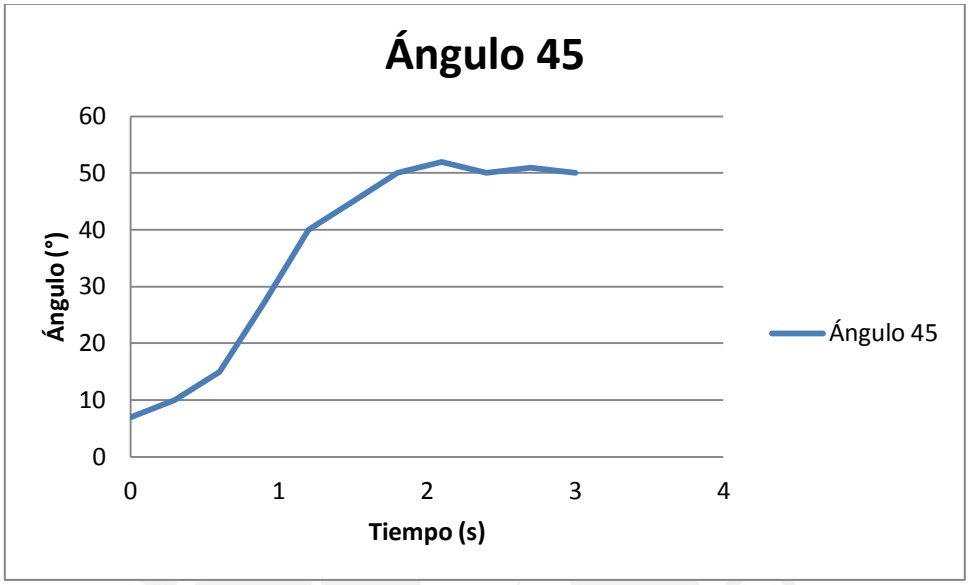

Figura 6: Kp=0.7, Ki=0.182, Kd=0.28. Fuente: Elaboración propia