SISTEMAS DE CONTROL

CONTROL DE BALANCÍN

Gonzalo Roldán, Bruno Fagni
UTN SAN FRANCISCO | INGENIERÍA ELECTRÓNICA
AÑO 2019

Índice

1. Introducción

Mediante el siguiente trabajo, se pretende explicar la implementación de un control de posición de una barra la cual puede girar libremente en una amplitud angular de 72º respecto a la horizontal. El sistema consiste en una barra cuyo giro se produce respecto a un eje que pasa por su centro de gravedad, el movimiento de giro será provocado por una fuerza de empuje producida por una hélice y un motor de tipo “Brushless”, de manera que controlando la velocidad de giro del motor se podrá regular la fuerza de empuje que actúa sobre la barra y con ello la posición de la misma.

2. Elementos necesarios

2.1 Modelo real de la planta

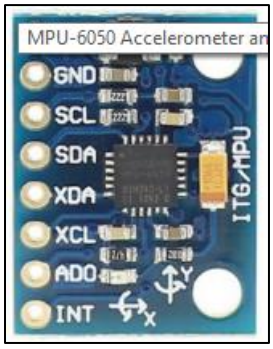
La planta como se observa en la figura está formada por una barra de madera, el balancín o barra móvil cuya longitud es de 90,8 cm y un doble soporte cuya altura es de 32,5 cm, el balancín va unido a cada soporte por medio de dos rulemanes que permiten el movimiento de giro, en uno de los extremos de la barra hay un soporte donde va colocado el motor brushless y la hélice, mientras que en el extremo contrario se encuentra fijado un contrapeso de madera.

Además de los componentes estructurales, es necesario el uso de diferentes dispositivos electrónicos. El circuito electrónico para hacer funcionar el motor se conoce como variador o ESC (Electronic Speed Controller), al cual se le envía una señal de información de PWM para controlar la velocidad de giro. Al ser un motor brushless, la fuente de alimentación va conectada al ESC y no directamente al motor, de forma que el ESC es el encargado de proporcionar la potencia necesaria al motor, en función de la señal de control que recibe.

El sensor en este caso es un acelerómetro con giróscopo incorporado, el mismo se conecta al microcontrolador por protocolo I2C y entrega valores correspondientes con una posición determinada del balancín.

2.2 Sensor acelerómetro y giróscopo

Para obtener la medida del ángulo en tiempo real, utilizaremos la unidad de movimiento inercial (IMU) MPU6050.



Es un dispositivo capaz de medir fuerza (aceleración) y velocidad, a partir de un acelerómetro y un giróscopo. No realiza una medición directa del ángulo, sino que requiere de ciertos cálculos para obtenerlo.

El MPU6050 opera con una tensión de 3.3 V ~ 5 V. Utiliza el protocolo de comunicación I2C.

2.2.1 Acelerómetro

Los datos de aceleración de cada uno de los tres ejes X, Y y Z, se guardarán en 2 registros de 8 bits cada uno.

Las ecuaciones para calcular los ángulos (brindadas por el fabricante) son:

$$AccAngle(Y) = \tan^{-1}\left(\frac{X}{\sqrt{Y^2 + Z^2}}\right)$$

$$AccAngle(X) = \tan^{-1}\left(\frac{Y}{\sqrt{X^2 + Z^2}}\right)$$

Como el ángulo es calculado respecto de la aceleración de la gravedad, no es posible calcular el ángulo respecto al eje Z con esta fórmula. Para ello se necesita un magnetómetro. Sin embargo, no es necesario para este trabajo.

2.2.2 Giróscopo

Este módulo es el encargado de medir la velocidad angular de nuestro dispositivo. Al igual que el acelerómetro, los datos de velocidad de cada eje se almacenan en 2 registros de 8 bits cada uno.

Las fórmulas para calcular los ángulos respecto de cada eje son:

$$GyroAngle(Y) = PreviousAngle(Y) + GyroData(Y) * elapsedTime$$

$$GyroAngle(X) = PreviousAngle(X) + GyroData(Y) * elapsedTime$$

Donde *elapsedTime* es el tiempo transcurrido cada vez que esta fórmula se calcula dentro del bucle del programa, *PreviousAngle* es el ángulo calculado en el bucle anterior y *GyroData* es el valor de la lectura de los registros de velocidad respecto de cada eje.

2.2.3 Filtrado y Reducción de Errores

El acelerómetro es capaz de medir cualquier ángulo. Sin embargo, sus lecturas son ruidosas y tienen cierto margen de error. Si además se agregan los datos del giróscopo el valor obtenido al final del cálculo no será preciso en ningún momento.

Por ello, es necesaria la utilización de filtros y combinar las lecturas de ambos módulos. Existen una gran variedad y cantidad de opciones. En este trabajo, simplemente usaremos un filtro complementario, que se adapta con gran certeza a nuestras necesidades.

El filtro complementario es una unión entre un filtro pasa altos para el giróscopo, y un filtro pasa bajos para el acelerómetro. La fórmula resultante es la siguiente.

$$TotalAngle(Y) = 0.98 * GyroAngle(Y) + 0.02 * AccAngle(Y)$$

$$TotalAngle(X) = 0.98 * GyroAngle(X) + 0.02 * AccAngle(X)$$

2.3 Motor Brushless

Este tipo de motor presenta buenas características y prestaciones para la construcción del prototipo Motor – Hélice – Balancín.

El motor Brushless es un motor eléctrico de corriente continua que no utiliza escobillas para el cambio de polaridad, sino que emplea un circuito electrónico externo conocido como variador electrónico o ESC (Electronic Speed Controller). En este tipo de motores, los imanes permanentes se encuentran en el rotor, mientras que el bobinado se encuentra en la parte fija o estator, lo contrario a como ocurre en los motores de corriente continua con escobillas.

El hecho de no usar escobillas hace que no sea necesario el rozamiento entre el rotor y la parte fija del motor, de tal forma que aumenta la eficiencia ya que se produce menor pérdida de calor, aumentando con ello el rendimiento con un menor consumo de potencia, y permite un rango de velocidad elevado al no tener limitaciones mecánicas. La ausencia de escobillas disminuye también el mantenimiento, así como el ruido electrónico.

Como desventaja se tiene un mayor costo de construcción, y la necesidad de un variador ESC para su control y funcionamiento, lo cual aumenta el precio del conjunto.

Al ser un tipo de motor muy común en el mundo del radiocontrol y el aeromodelismo, se dispone de una gran variedad tanto de motores brushless como de ESC en el mercado, con diferentes características y precios.

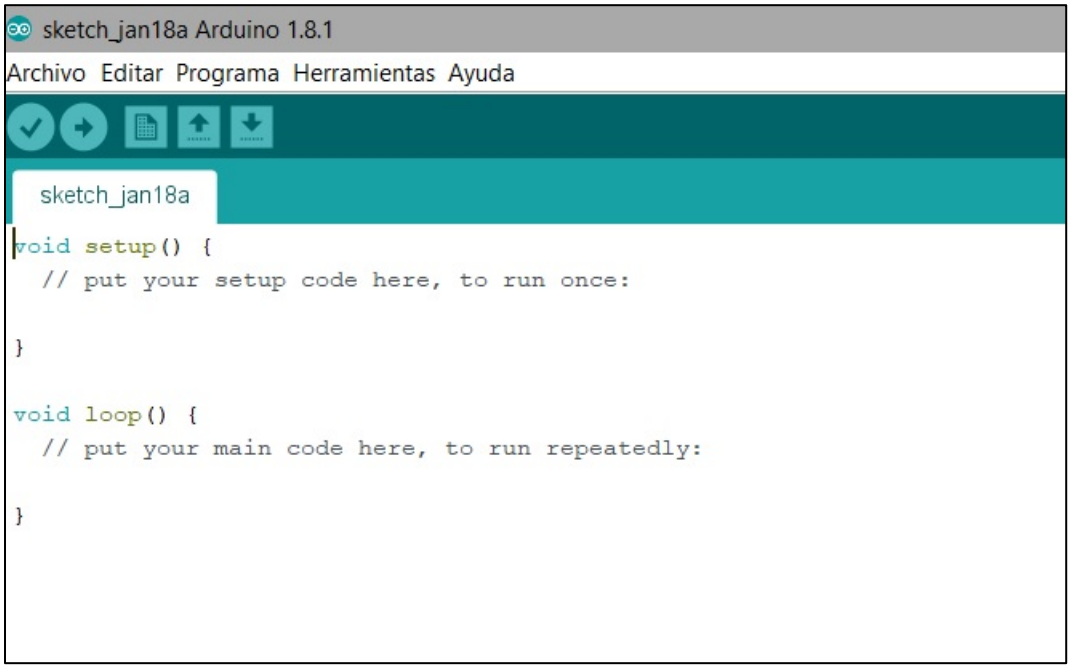
2.4 Variador Electrónico (ESC)

Como se ha comentado en los apartados anteriores, los motores brushless requieren de una electrónica de control que les proporcione la señal de potencia. Esta electrónica de control se conoce como variador electrónico o ESC, que transforma la corriente continua en una tensión alterna y es el encargado de aplicar al motor la señal necesaria para ajustar las revoluciones del motor, es decir, permite controlar la velocidad del motor.

Un ESC suele contar con siete conexiones necesarias para el correcto funcionamiento del motor: tres cables de salida que van conectados directamente al motor y son los que le proporcionan la señal de control, dos cables de alimentación usados para conectar la fuente de alimentación o batería (positivo y negativo), un cable de señal de información por donde recibe la señal PWM de control de velocidad del motor, y otro cable para la toma de tierra.

2.5 Arduino

Arduino es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexibles y fáciles de usar. El lenguaje de programación Arduino se basa en C/C++. En general, un programa en Arduino tiene la estructura que se observa en la siguiente figura.



En donde **setup()** es la parte encargada de recoger la configuración , se invoca una sola vez cuando el programa empieza. Se utiliza para inicializar los modos de trabajo de los pines, o el puerto serie. Debe ser incluida en un programa, aunque no haya declaración que ejecutar.

La función **loop()** contiene el código que se ejecutara cíclicamente lo que posibilita que el programa este respondiendo continuamente ante los eventos que se produzcan en la tarjeta (lectura de entradas, activación de salidas, etc.). Esta función es el núcleo de todos los programas de Arduino y la que realiza la mayor parte del trabajo.

Ambas funciones son necesarias para que el programa trabaje.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede incidir sobre aquello que le rodea controlando luces, motores y otros actuadores.

3. Programación del Control

El funcionamiento básico del programa consiste en realizar una lectura de los datos brindados por el sensor MPU6050. A partir de los mismos calculamos la inclinación real de los ejes. Utilizaremos solamente la inclinación del eje Y para el control, calculando el ángulo de giro respecto de dicho eje.

El ángulo de comparación (SetPoint) será el de 0°, es decir el balancín debe estar en posición horizontal. Sin embargo, este valor podrá ser fácilmente modificable.

Para controlar el motor Brushless enviamos una señal PWM al variador ESC con un pulso entre 1000us y 2000us.

3.1 Librerías

En primer lugar, definimos las librerías necesarias para la comunicación I2C, el control del motor y el cálculo del PID.

```
#include <Wire.h>
#include <Servo.h>
#include <PID_v1.h>
```

Luego se declaran todas las variables y constantes para guardar los datos de Acelerómetro y Giróscopo, y las utilizadas para el control PID.

3.2 Setup

En el *setup*, se configura la comunicación I2C con el módulo MPU 6050 y se establece el tiempo de muestreo del PID, el valor del Setpoint y los límites de la salida del sistema.

3.3 Loop

Definimos la secuencia que se ejecutará constantemente (*loop*).

3.3.1 Calibración del ESC

Se lleva a cabo la calibración del variador, donde se establecen los valores de velocidad mínimo y máximo, rutina necesaria para el funcionamiento del mismo.

```
void calibrar_ESC()
{
  while (countFlag < 2) {
    if ((countFlag == 0) && (mensaje1 == 0))
    {
      Serial.println("Ingrese velocidad maxima y luego encienda la fuente
de alimentacion");
      mensaje1++;
    }
    else if ((countFlag == 1) && (mensaje2 == 0))
    {
      Serial.println("Ingrese velocidad minima");
      mensaje2++;
    }
    if (Serial.available() > 0)
    {
      palabra = Serial.readString();

      velocidad = palabra.toInt();

      if ((velocidad <= 2000) && (velocidad >= 1000))
      {

        motor_prop.writeMicroseconds(velocidad);

        Serial.println(velocidad);
        countFlag++;
        delay(2000);
      }
    }
  }
  motor_prop.writeMicroseconds(1100);
  delay(2000);
}
```

3.3.2 Lectura y Tratamiento de Datos del Sensor MPU6050

Obtenemos los datos brindados por el sensor, realizando una lectura de los 6 registros de acelerómetro y 4 de giróscopo. Lo primero que hacemos es calcular el tiempo transcurrido entre un bucle y otro, dato necesario para obtener el ángulo a partir de los valores de los registros del giróscopo.

```
void CalculoMPU6050() {

  elapsedTime = (millis() - timePrev) / 1000;   //tiempo transcurrido
  timePrev = millis();
```

Comenzamos con la lectura de los datos, en primer lugar, leemos los registros del Acelerómetro y luego los del Giróscopo.

```
 //----------- LECTURA DE DATOS DEL ACELEROMETRO -----------//

  //--- Registros de Acelerometro ---//

  AcX = Wire.read() << 8 | Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C
(ACCEL_XOUT_L), lectura del byte superior e inferior
  AcY = Wire.read() << 8 | Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E
(ACCEL_YOUT_L)
  AcZ = Wire.read() << 8 | Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40
(ACCEL_ZOUT_L)

  //---------- CÁLCULO DEL ÁNGULO A PARTIR DEL ACELEROMETRO --------//

  Acc_AngleY = atan(-1 * (AcX_G) / sqrt(pow((AcY_G), 2) + pow((AcZ_G),
2))) * rad_to_deg; //valor del angulo respecto del eje Y

  //---------- LECTURA DE DATOS DEL GIROSCOPO -----------//

  //--- Registros de Giroscopo ---//

 GyX = Wire.read() << 8 | Wire.read();

 GyY = Wire.read() << 8 | Wire.read();

  //----------- CÁLCULO DEL ÁNGULO A PARTIR DEL GIROSCOPO -----------//
  /*Para obtener los datos del giroscopo en grados por segundo, dividimos
el valor por 131 (brindado por datasheet) */

  float GyX_D = GyX / 131.0; //Datos giroscopo en grados por segundo
  float GyY_D = GyY / 131.0; //Datos giroscopo en grados por segundo

 Gy_AngleY = GyY_D * elapsedTime;
```

Por último, obtenemos el ángulo total respecto del eje Y, a partir de la ecuación del filtro complementario explicada con anterioridad.

```
  //------------ ANGULO TOTAL RESPECTO EJE Y ------------//

 Total_Angle = 0.98 * (Total_Angle + Gy_AngleY) + 0.02 * Acc_AngleY;
```

3.3.3 Cálculo del Control PID

Las constantes obtenidas luego de numerosas pruebas son las siguientes:

```
double consKp = 5.00, consKi = 25.39, consKd = 2.28;
```

Se establece el valor del ángulo como la entrada al sistema y se establecen las constantes de sintonización anteriores.

```
//-------------- CALCULO DEL PID -------------------//
Input = Angulo;
ControlPID.SetTunings(consKp, consKi, consKd);
```

Para que el cálculo del PID sea lo más eficaz posible, debemos ejecutar el cómputo en un período de muestreo constante, igual al tiempo de muestreo. Es decir, no se debe calcular el PID en aquellos casos en que el tiempo de ejecución del programa modifique dicho tiempo.

```
time_now = millis();
tiempo_transcurrido = time_now - tiempo_previo;

if (tiempo_transcurrido == 21)
{
  ControlPID.Compute();
}
```

Finalmente, enviamos los datos de salida del PID al motor Brushless.

```
motor_prop.writeMicroseconds(Output);
```

Luego de un gran número de pruebas con diferentes constantes de sintonización, se observó que el sistema se estabilizaba dentro de un rango cercano al valor de Setpoint, pero cada cierto tiempo aleatorio se producía un decremento pronunciado de velocidad del motor. Esto se debió a que el sensor presenta gran cantidad de ruido ante vibraciones de la estructura. Por eso, fue necesario realizar un promedio de diez muestras del ángulo obtenido, previo al cálculo del control PID.

```
for (i = 0; i < N ; i++)
{
  CalculoMPU6050();
  medida_acumulada = medida_acumulada + Total_Angle;
}
double Angulo = medida_acumulada / N;
```

Se pudo visualizar una mejor de la respuesta del sistema, manteniéndose la mayoría del tiempo entre 0° u -1° con oscilaciones aleatorias de 2 a 4 grados alejadas del Setpoint, obteniendo un error del 4,3% como máximo.

A pesar de realizar un promedio de las muestras, el sensor sigue presentando ruido ante vibraciones del motor, a lo cual le atribuimos estas pequeñas oscilaciones.

4. Modelado matemático

4.1 Movimiento de rotación de un sólido rígido

El movimiento a controlar en esta planta es el de rotación de un sólido alrededor de su eje. La variación del estado de rotación de un sólido viene determinada por la variación de su velocidad angular por lo que, si queremos describir el movimiento de rotación debemos encontrar una ecuación que nos permita calcular la aceleración angular del mismo, tal como la siguiente ecuación:

$$\sum \vec{r}\ x\ \vec{F}ext = I\vec{\alpha}$$

Ésta es la ecuación del movimiento de rotación de un sólido rígido (segunda ley de Newton) y será la empleada para plantear la ecuación que rige el movimiento de nuestro sistema.

4.2 Momento de inercia

4.2.1 Momento de inercia para una masa puntual

En este caso el sistema contiene tres elementos que aportan momento de inercia, uno es la masa de la barra, otro la masa del motor y por último la masa correspondiente al contrapeso. Cada uno de estos aporta un momento de inercia que se calcula de manera independiente y que finalmente sumaremos para caracterizar el momento de inercia del sistema en general.

$$\sum m_i l_i^2$$

Siendo:

- $mi: masa\ puntual\ del\ objeto.$
- $li: distancia\ del\ eje\ de\ rotación\ a\ la\ masa\ del\ objeto.$

4.2.2 Calculo del momento de inercia producido por el motor

Masa del motor: 66 gr.

Distancia del motor al eje de giro: 45,4 cm.

Entonces:

$$I_m = m_m l_m^2 = 0.066 * 0.454^2 = 0.0272\ kg * m^2$$

4.2.3 Calculo del momento de inercia producido por el contrapeso

Masa del contrapeso: 29 gr.

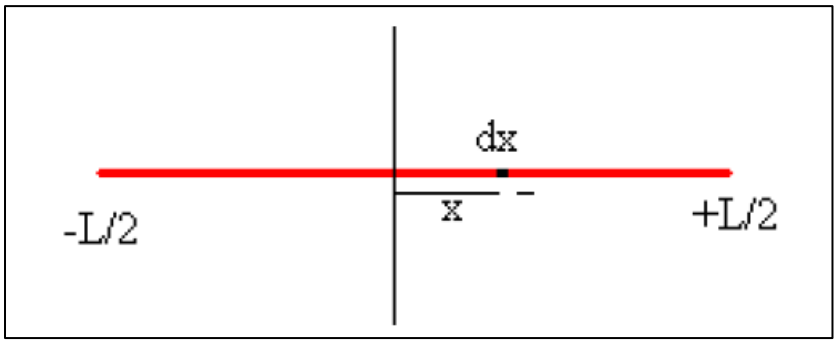
Distancia del contrapeso al eje de giro: 45,4 cm.

Entonces:

$$I_c = m_c l_c^2 = 0.029 * 0.454^2 = 0.0119\ kg * m^2$$

4.2.4 Momento de inercia de una barra con eje de giro en su centro de gravedad

Apoyándonos en el siguiente gráfico vamos a calcular el momento de inercia de una barra de masa M y longitud L respecto de un eje de giro perpendicular a la longitud de la barra que pasa por el centro de gravedad.



La masa dm del elemento de longitud de la varilla comprendido entre x y x+dx es:

$$dm = \frac{M}{L}\ dx$$

Por ende, el momento de inercia de la barra es:

$$I_v = \int_{-\frac{L}{2}}^{+\frac{L}{2}} \frac{M}{L} * x^2\ dx = \frac{1}{12} ML^2$$

4.2.5 Calculo del momento de inercia producido por la barra

Masa de la barra: 230 gr.

Longitud de la barra: 90,8 cm.

Entonces:

$$I_v = \frac{1}{12} ML^2 = \frac{1}{12} 0.230 * 0.908^2 = 0.0158\ kg * m^2$$

4.2.6 Momento de inercia total

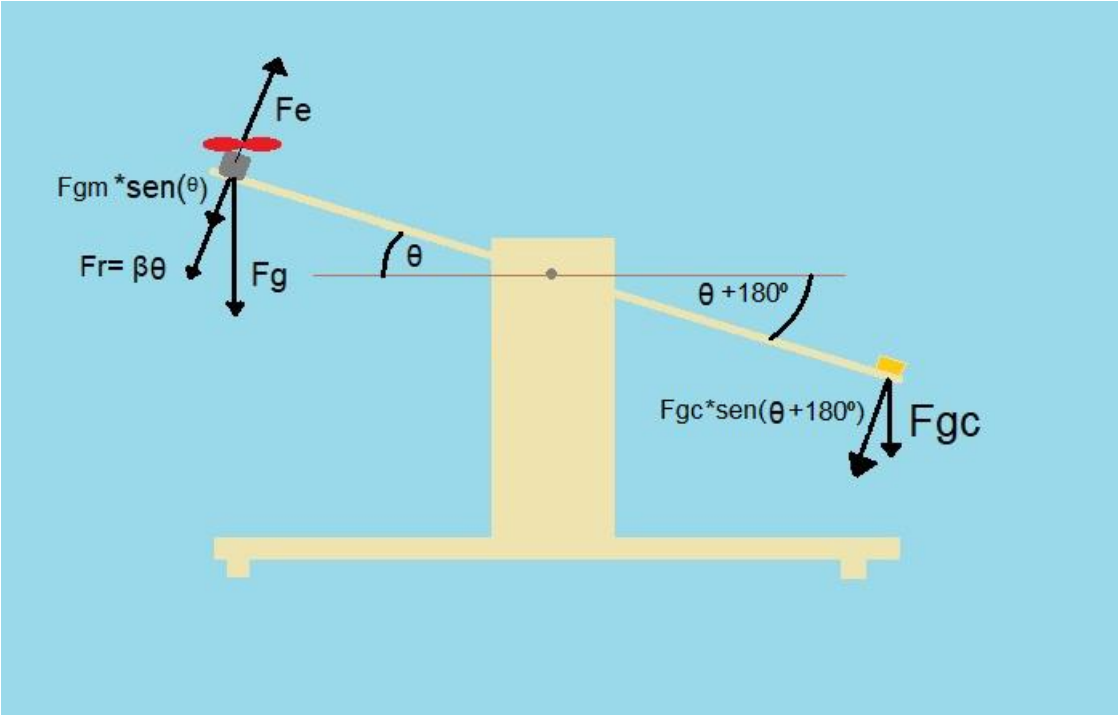
El momento de inercia total se obtiene realizando la sumatoria de cada momento individual aportado al sistema.

$$I_t = I_m + I_c + I_v = 0.0272 + 0.0119 + 0.0158 = 0.0311\ kg * m^2$$

4.3 Obtención de la función de transferencia

La relación que se quiere obtener es como varia la posición θ en función de la fuerza de empuje producida por la hélice Fe. Para ello se busca la relación de equilibrio de fuerzas del sistema.

A continuación, se realiza un bosquejo del sistema real.



Para determinar el modelo matemático aproximado de este sistema, se empleará la ecuación del movimiento de rotación de un sólido (segunda ley de Newton). Una vez aplicada esta ecuación y luego de haber obtenido cada uno de sus parámetros se realizará la transformada de Laplace a la ecuación resultante, así de este modo se podrá obtener la función de transferencia del modelo aproximado, sobre la cual se desarrollarán algunas pruebas para comprobar la respuesta del sistema modelado y compararla con la del sistema real.

Para obtener la ecuación de movimiento que caracteriza a la planta se tendrá en cuenta que el eje de inercia de la barra móvil está ubicado justo en su centro de gravedad, por ende, la fuerza resultante sobre la barra será nula, al estar todas las fuerzas de un lado y del otro, de su eje de inercia, compensadas entre sí.

Cabe señalar que solo serán de interés las fuerzas debidas a la masa del motor, las de fricción con el eje de giro y la masa del contrapeso.

4.3.1 Ecuación de modelado aproximado del sistema

$$LF_e + LFg_c * \sin(180º + \theta) - LFg_m * \sin(\theta) - L\beta * \sin(\theta) = I\alpha$$

Para pequeñas variaciones de ángulos consideramos sen(θ) = θ, quedando entonces:

$$LF_e + LFg_c * (-\theta) - LFg_m * (\theta) - L\beta * (\theta) = I\alpha$$ *Notar que θ+180 = - θ.*

Reacomodando y despejando α nos queda:

$$\frac{LF_e}{I} - \frac{LFg_c * (\theta)}{I} - \frac{LFg_m * (\theta)}{I} - \frac{L\beta * (\theta)}{I} = \alpha$$

Aplicando la trasformada de Laplace:

$$\frac{LF_e(s)}{I} - \frac{LFg_c * \theta(s)}{I} - \frac{LFg_m * \theta(s)}{I} - \frac{L\beta * S\theta(s)}{I} = S^2\theta(s)$$

Despejando:

$$\frac{LF_e(s)}{I} = S^2\theta(s) + \frac{L\beta * S\theta(s)}{I} + \theta(s) * \left(\frac{LFg_m}{I} + \frac{LFg_c}{I}\right)$$

Factorizando:

$$\frac{L}{I}F_e(s) = \theta(s) * \left[S^2 + \frac{L}{I}\beta S + \frac{L}{I} * (Fg_m + Fg_c)\right]$$

Finalmente reacomodamos para dejar la salida del sistema sobre la entrada:

$$\frac{\theta(s)}{F_e(s)} = \frac{{}^{L}/_{I}}{S^2 + \frac{L}{I}\beta S + \frac{L}{I}(Fg_m + Fg_c)}$$

Siendo:

- $L = Distancia\ al\ eje\ de\ rotación.$
- $I = Momento\ total\ de\ inercia.$
- $Fgm = Fuerza\ gravitatoria\ del\ motor.$
- $Fgc = fuerza\ gravitatoria\ del\ contrapeso.$
- $\beta = Coeficiente\ de\ rozamiento.$

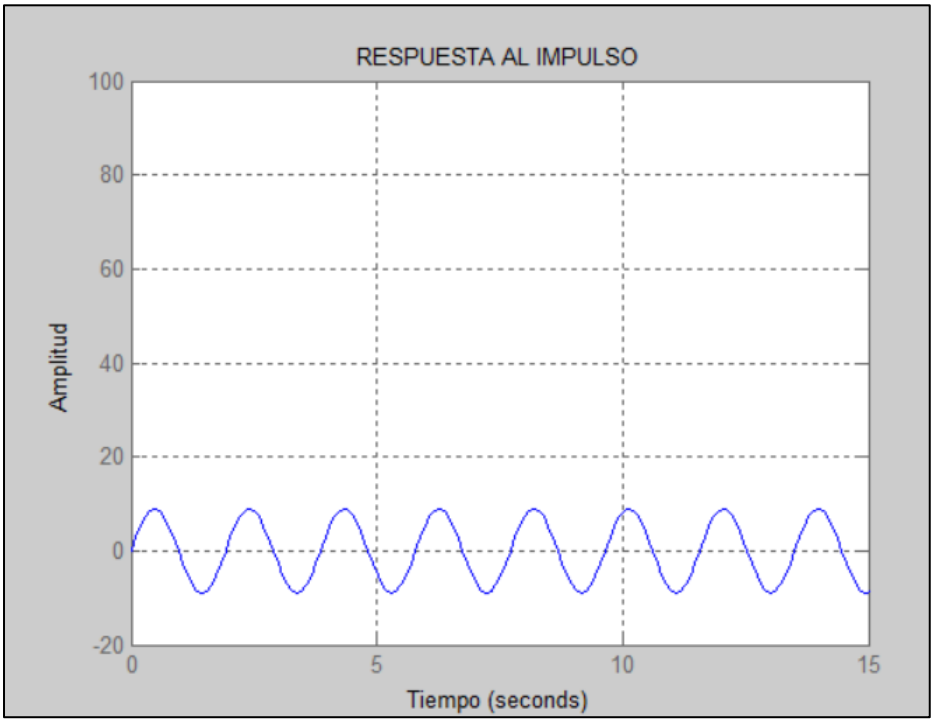
Nota: a modo de simplificación de cálculos se consideró β=0 ya que para nuestra aplicación podemos considerar despreciable el coeficiente de rozamiento de los cojinetes.

4.4 Sintonización y visualización de la respuesta del sistema modelado

Una vez obtenida la ecuación de la planta en el plano complejo “S” procedemos a introducir los valores en un Script de Matlab y así graficar las diferentes respuestas del sistema.

Las siguientes imágenes detallan gráficamente las características y la respuesta el sistema modelado.

4.5 Respuesta al impulso del sistema



4.6 Respuesta a lazo cerrado del sistema

Haciendo un análisis visual de nuestra función de transferencia podemos notar que se asemeja a la ecuación estándar de un sistema de segundo orden:

$$\frac{Y(s)}{X(s)} = \frac{K\omega_n^2}{S^2 + 2\zeta\omega_n S + \omega_n^2}$$

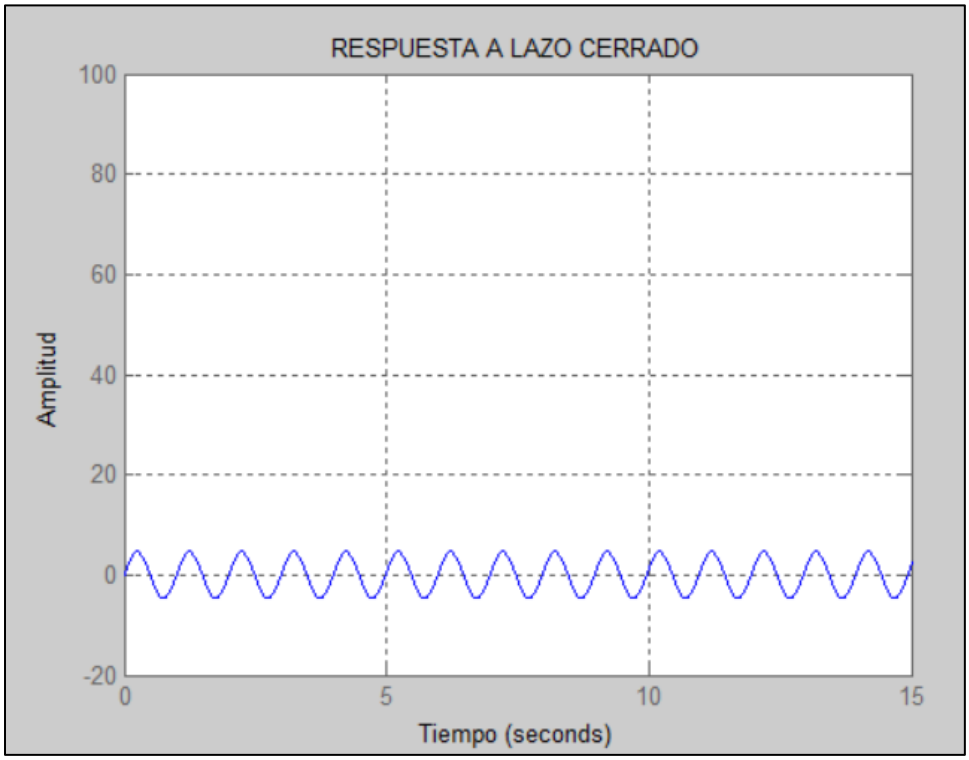
Puede verse la semejanza con nuestro sistema:

$$\frac{\theta(s)}{F_e(s)} = \frac{{}^{L}/_{I}}{S^2 + \frac{L}{I}\beta S + \frac{L}{I}(Fg_m + Fg_c)}$$

Como se mencionó anteriormente, se consideró despreciable el coeficiente de rozamiento β, por ende, nuestro termino lineal quedaría como sigue:

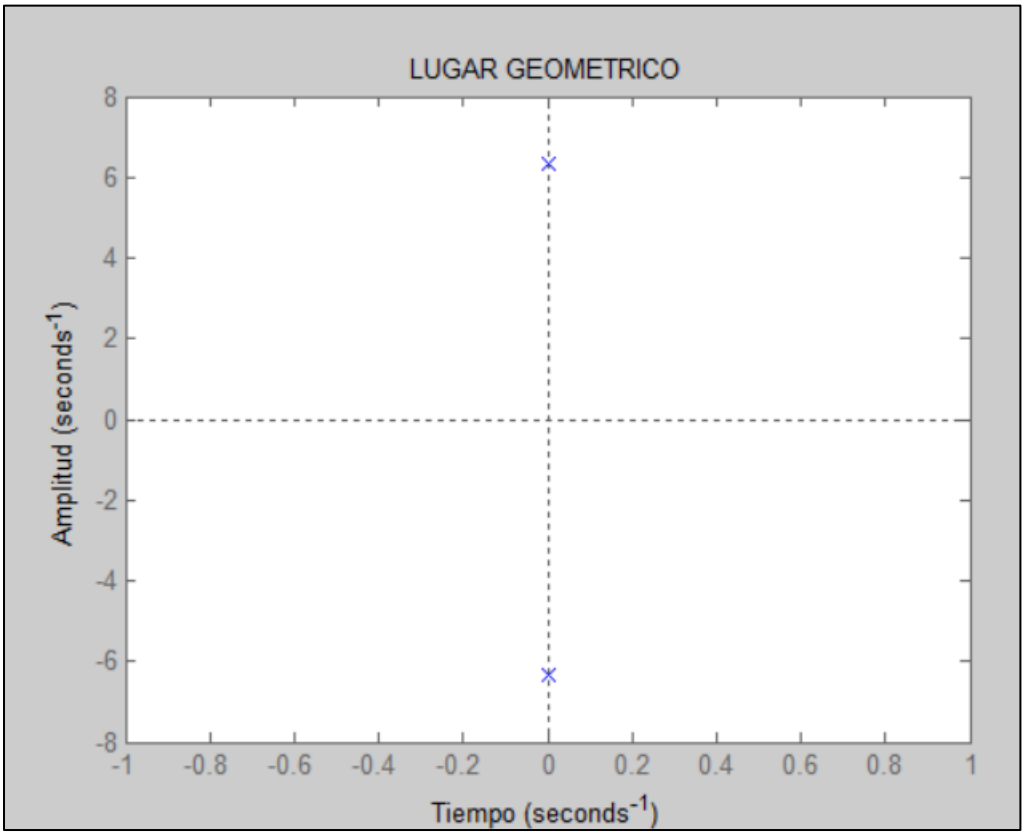
$$2\sqrt{\frac{L}{I}(Fg_m + Fg_c)} * \zeta = 0$$

Puede deducirse entonces que para nuestro sistema aproximado se considera que el coeficiente de amortiguamiento ζ es nulo. Esto justifica las oscilaciones constantes en el siguiente grafico que corresponde a la respuesta a lazo cerrado de nuestro sistema.



4.7 Lugar geométrico de los polos y ceros del sistema

En el análisis del lugar geométrico de los polos de nuestro sistema a lazo cerrado podemos observar que el sistema es críticamente estable, ya que posee 2 polos ubicados sobre el eje imaginario, esto se debe a que nuestro sistema posee una respuesta finita distinta de cero al impulso en un tiempo considerado infinito.



4.8 Respuesta del sistema al control PID

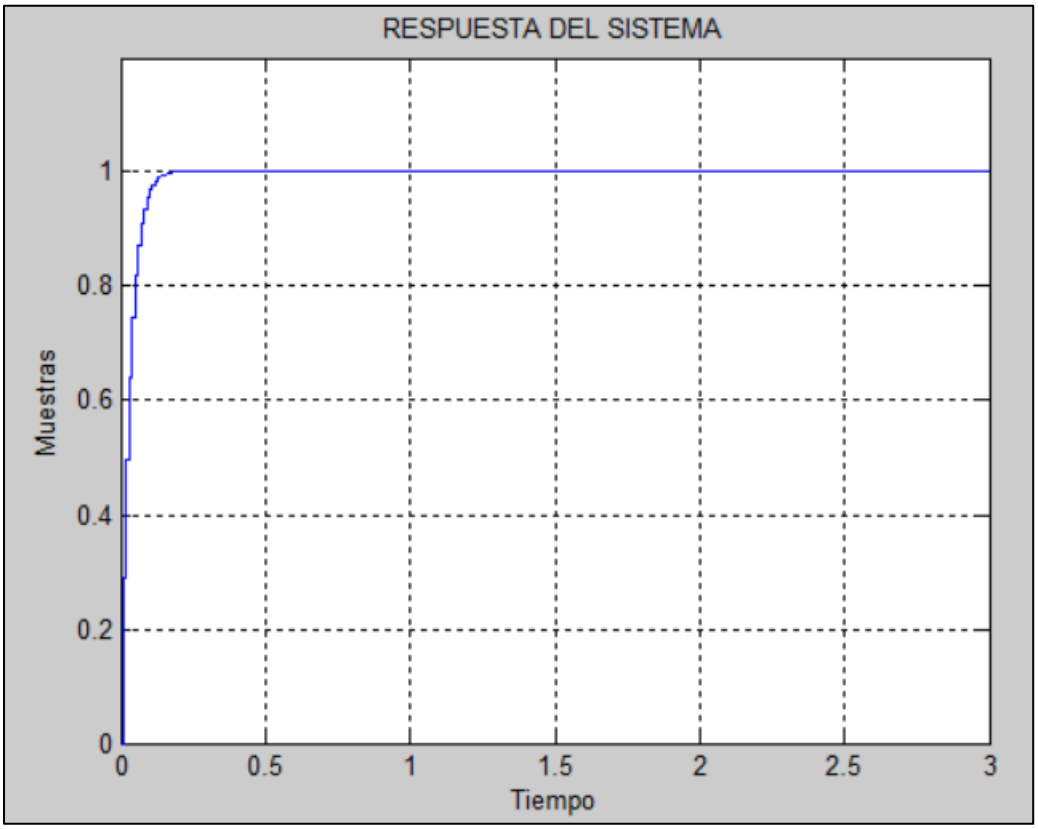
Para sintonizar el control PID y así lograr la mejor respuesta del sistema, se utilizó un método de prueba y error, comenzando por un control meramente proporcional hasta que la curva respuesta del sistema se aproximó a la respuesta deseada. Luego se procedió a anular las grandes elongaciones y mejorar el tiempo de crecimiento adicionando parte de un control derivativo, obteniendo como resultado una respuesta oscilante constante con un error de estado estacionario pequeño. Por último, se adicionó un control integrativo a fin de reducir las oscilaciones en el estado estacionario. Al final del proceso se debió modificar un poco el conjunto derivativo e integrativo para hallar la respuesta deseada en el menor tiempo posible.

Dicho proceso nos arrojó como resultado el siguiente conjunto de constantes:

$$Kp = 1.51$$

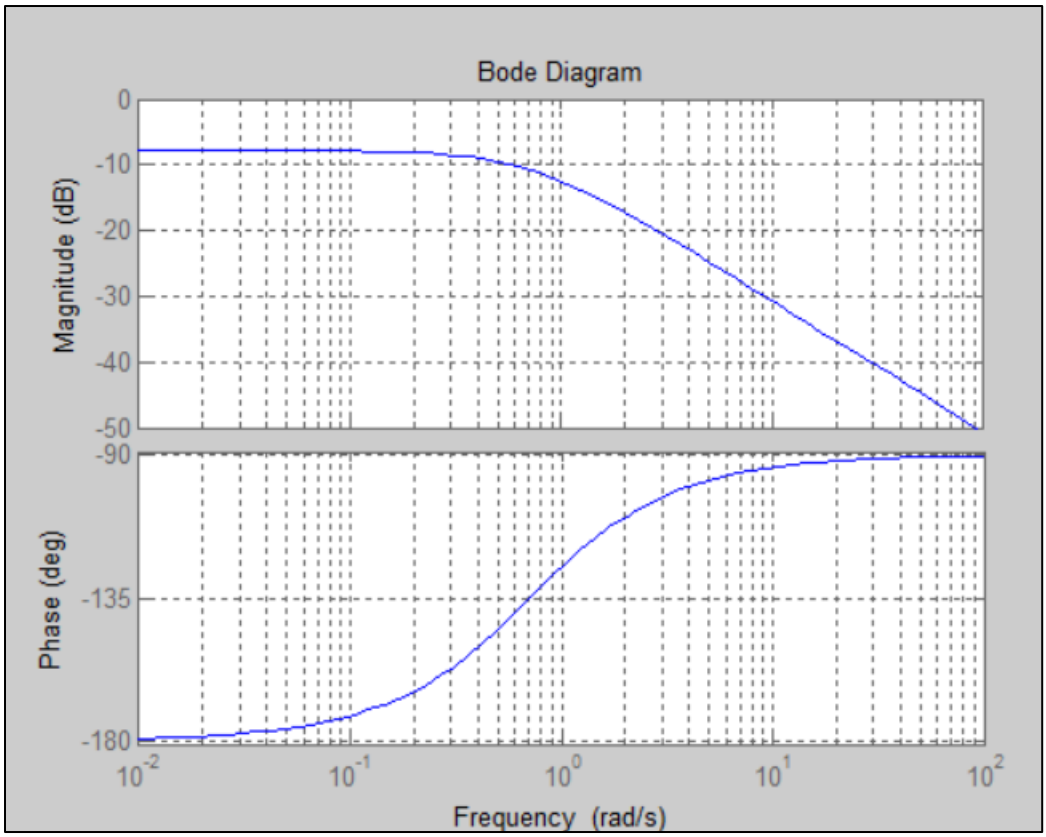
$$Kd = 4.95$$

$$Ki = 55$$



4.9 Diagrama de bode del comportamiento del sistema:

Mediante el diagrama de bode de la respuesta del sistema a lazo cerrado, podemos observar que para cualquier señal de entrada del sistema nuestra salida se va a ver atenuada y en cuanto al diagrama de fase, podemos observar que la salida del sistema a frecuencias bajas presenta un mayor atraso respecto a la entrada que en frecuencias más elevadas.



5. Conclusión

A modo de conclusión, podemos destacar la importancia de este proyecto para lograr entender el funcionamiento del control PID de una planta y poder aplicarlo luego en el ámbito industrial. Al ser un tema tan complejo, se simplificó en gran medida la teoría, pudiendo aplicar varios de los conocimientos adquiridos en clases.

En lo que respecta al resultado final del proyecto, se logró obtener una ecuación de transferencia que represente correctamente la planta y se evaluaron las diferentes condiciones de sintonización, obteniendo las 3 constantes finales como mejor acción de control en el marco teórico.

En la práctica, nos resultó muy complejo poder llegar a un conjunto de constantes para estabilizar el sistema en un valor cercano al Setpoint. Luego de innumerables intentos, se obtuvieron una serie de valores, descriptos con anterioridad, con los cuales se alcanzó un error máximo del 4%, es decir, en desplazamiento angular total del balancín de 72°, el error máximo fue de 4°, manteniéndose la mayoría del tiempo en 0° con oscilaciones aleatorias de valor menor a 2°.

Atribuimos este problema al ruido provocado por las vibraciones del motor, que ocasiona una lectura errónea del sensor.

6. Bibliografía

- http://brettbeauregard.com/blog/2017/06/introducing-proportional-on-measurement/
- http://brettbeauregard.com/blog/wp-content/uploads/2012/07/Gu%C3%ADa-de-uso-PID-para-Arduino.pdf
- http://www.sinaptec.alomar.com.ar/2017/10/tutorial-23-esp8266-obtener-inclinacion.html
- http://kio4.com/arduino/46giroscopio.htm
- https://dronesandrovs.wordpress.com/2012/11/24/how-to-control-a-brushless-motor-esc-with-arduino/
- https://www.monografias.com/trabajos106/control-pid-y-su-aplicacion-automatizacion/control-pid-y-su-aplicacion-automatizacion2.shtml
- https://controlautomaticoeducacion.com/microcontroladores-pic/19-control-pid-en-pic-ejemplo-2/