

# Explicación del Controlador ArduinoQuadcopter

Ing. Aldric Ian López Vergara Anaya

**Abstract**—Este pequeño artículo tiene la finalidad de documentar y explicar el funcionamiento del proyecto ArduinoQuadcopter. Aunque también funciona para otros quads.

**Index Terms**—Control, PID, Acelerómetro, Magnetómetro, Barómetro, Giroscopio.

## I. INTRODUCCIÓN

EN este artículo se explicara brevemente el funcionamiento del programa para el Arduino Quadcopter. Se explican conceptos de control necesarios así como el uso de distintos sensores del IMU para obtener la posición del sistema. Abril 19, 2015

## II. PROGRAMA ARDUINOQUADCOPTER

### A. Sistema de Control

El sistema de control es la parte más importante para que el multirrotor pueda volar. A grandes rasgos, el sistema de control son una serie de ecuaciones matemáticas que procesa en este caso una pequeña computadora, llamada microcontrolador, para mantener al vehículo en un vuelo estable. Cuando el multirrotor se encuentra en vuelo, está sometido a muchas fuerzas externas, llamadas perturbaciones, debido al viento, imperfecciones de construcción, etc. Estas perturbaciones pueden hacer que el multirrotor no se mantenga estable en el aire. Por ejemplo si nosotros queremos que se mantenga completamente horizontal pero debido a la viento en realidad se termina inclinando levemente, este ángulo de inclinación es lo que se conoce en control como error. En general este error se expresa como una simple resta entre el valor deseado y el valor real de la siguiente manera:

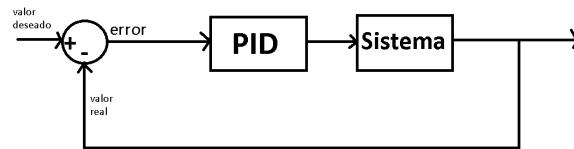
$$e(t) = x_{deseado}(t) - x_{real}(t)$$

En general, el sistema de control intenta minimizar el valor de  $e(t)$  para el menor tiempo  $t$  posible. Existen varias estrategias que se pueden tomar para diseñar estas ecuaciones que se conocen como algoritmos de control. Dependiendo de la complejidad del sistema que se quiere controlar, en este caso el multirrotor, así como de la capacidad de la computadora que se va a utilizar, se elige dicho algoritmo. En el caso de los multirrotos, el algoritmo de control por excelencia es el llamado **PID**.

1) *¿Qué es un PID?*: Un **PID** es un algoritmo de control que combina tres formas del error que se quiere minimizar (la parte proporcional, la integral y la derivativa, por ende el nombre de PID). El **PID** hace esto mediante una suma de estas tres formas de error asignándole un peso específico a cada una por medio de tres constantes ( $k_p$ ,  $k_i$ ,  $k_d$  respectivamente).

$$PID(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{d}{dt} e(t)$$

Este valor que se obtiene para  $PID(t)$  se le envía al sistema, en este caso al multirrotor, por medio de una señal eléctrica. El sistema cambia entonces debido a esta señal lo que modifica a su vez el error. El nuevo error es calculado y la computadora se encarga de obtener un nuevo valor para el **PID** utilizando la ecuación anterior, completando así el ciclo. Generalmente este tipo de controles se expresa mediante un diagrama de bloques:



2) *¿Como se aplica el PID al multirrotor?*: En el multirrotor contamos con actuadores (motores), sensores (acelerómetro, giroscopio, magnetómetro, barómetro, gps, etc), y un microcontrolador (computadora). Para implementar el control se debe conectar el microcontrolador a los actuadores y sensores. Los sensores le envían al microcontrolador señales eléctricas que representan variables físicas y le permiten encontrar el valor real para calcular el error. Por otro lado los actuadores son la forma que el microcontrolador puede interactuar con el sistema, y son a los que se envía el valor del PID. Lo primero que se hace es definir las variables que se quieren controlar, en el caso del multirrotor se pueden controlar 3 ángulos diferentes, así como 3 direcciones espaciales. En el caso del multirrotor es indispensable que la computadora se encargue de controlar los tres ángulos, ya que un piloto humano no sería capaz de reaccionar lo suficientemente rápido como para corregir los errores. Por otro lado las tres direcciones pueden ser controladas por el piloto ó en el caso de un multirrotor completamente autónomo, mediante controladores **PID** similares a los que utilicen GPS y barómetro para determinar una posición exacta deseada. En este caso nos enfocaremos en discutir cómo controlar los tres ángulos de giro, ya que una vez controlados es mucho más sencillo implementar un control para el resto de las direcciones.

### B. Midiendo las Variables

Ahora que sabemos que se van a controlar los tres ángulos, es importante saber cómo medirlos para poder calcular el

error. Existen 3 sensores que, combinados, nos permiten hacer dichas mediciones. Estos son: el acelerómetro, giroscopio, y magnetómetro. El acelerómetro es un sensor que mide la aceleración neta del multirrotor (utilizando una constante acorde al sensor para convertir la señal eléctrica a  $m/s^2$ ). Si bien parece no ser muy útil, cuando el multirrotor no se está moviendo, o se mueve a una velocidad estable, la única aceleración a la que está sometido es debido a la gravedad. Como sabemos que la gravedad es siempre en dirección vertical hacia abajo, es posible calcular mediante trigonometría el ángulo en el que se encuentra el multirrotor:

$$\Theta = -\arctan(\text{acc}_x \text{acc}_y^2 + \text{acc}_z^2)$$

$$\Gamma = -\arctan(\text{acc}_y \text{acc}_x^2 + \text{acc}_z^2)$$

Donde  $\Theta$  es el ángulo hacia  $x$  y  $\Gamma$  el ángulo hacia  $y$ . Para obtener el otro ángulo, se utiliza otro sensor llamado magnetómetro. El magnetómetro mide el campo magnético terrestre en las direcciones  $x, y, z$ . Si suponemos que el multirrotor se encuentra horizontal, el campo magnético únicamente está en los ejes  $x, y$ . De esta forma, utilizando trigonometría nuevamente, se puede conocer, al igual que una brújula, la dirección del norte; y por ende, el ángulo al que está orientado el multirrotor.

$$\Phi = \arctan\left(\frac{my}{mx}\right)$$

Por último, el giroscopio es un sensor un poco diferente; ya que lo que nos permite medir es la velocidad angular ( $\omega$ ). Es decir, que tan rápido o que tan lento está girando sobre alguno de estos ejes. Sin embargo, es posible obtener el ángulo a partir de esta medición; debido a que la velocidad angular es igual a la derivada con respecto al tiempo del mismo ángulo.

$$\omega = \frac{d}{dt} \Theta$$

Despejando el ángulo:

$$\Theta = \int \omega dt$$

Sin embargo, como los microcontroladores que se usan para este tipo de aplicaciones no pueden hacer ni integrales ni derivadas, es necesario aproximar esos valores mediante operaciones más sencillas. En el caso de la integral, se aproxima utilizando el método de los rectángulos; que no es más que una multiplicación del tiempo entre dos mediciones, sumada con los valores de ángulo anteriores.

$$\Theta = \Theta_{anterior} + (t_2 - t_1)\omega$$

Esto se puede aplicar para los tres ángulos respectivamente, por lo que ahora se tienen dos mediciones de sensores diferentes para cada ángulo.

Finalmente, se deben combinar estas mediciones. Es importante juntar dos sensores para cada uno de los ángulos, ya que cada sensor tiene una mayor o menor confiabilidad dependiendo de las condiciones. En el caso del acelerómetro y del magnetómetro, funcionan mejor cuando la velocidad angular es cercana a cero; es decir, que los ángulos no cambian

muy rápidamente. Por otro lado, el giroscopio, al ser un medidor de velocidades angulares, funciona mejor cuando los ángulos cambian rápidamente; pero muy malo para cuando las velocidades angulares son cercanas a cero. Los sensores se combinan entonces utilizando un filtro complementario de la siguiente forma:

$$\Theta = \alpha(\Theta_{anterior} + \omega_{\Theta_{giro}}(t_2 - t_1)) + (1 - \alpha)\Theta_{acel}$$

$$\Gamma = \alpha(\Gamma_{anterior} + \omega_{\Gamma_{giro}}(t_2 - t_1)) + (1 - \alpha)\Gamma_{acel}$$

$$\Phi = \alpha(\Phi_{anterior} + \omega_{\Phi_{giro}}(t_2 - t_1)) + (1 - \alpha)\Phi_{acel}$$

Donde  $\alpha$  es una constante entre 0 y 1 que nos indica la confiabilidad que se tiene para cada medición. Esta constante depende de la combinación de sensores que se utiliza para cada medición. En este caso es preferible tener una con valores altos (usualmente entre 0.85 y 0.99) de forma tal que se confíe más en el giroscopio cuando haya velocidades angulares grandes y menos cuando sean pequeñas. Estos valores para los tres ángulos son los que ahora se envían como retroalimentación de valor real para calcular el error del PID.

### C. Implementando el PID

La ecuación del **PID** es una ecuación integro-diferencial; es decir, que incluye tanto integrales como derivadas. Por lo que es necesario reemplazarlas por aproximaciones discretas. En este caso se pueden utilizar aproximación de rectángulos para las integrales, y diferencias para las derivadas. Por lo que la ecuación de PID:

$$PID(t) = k_p e(t) + k_i \int e(t) dt + k_d \frac{d}{dt} e(t)$$

Se convierte en:

$$PID(n) = k_p e(n) + k_i e(n)(t_n - t_{n-1}) + I_{anterior} + k_d \frac{e(n) - e(n-1)}{t_n - t_{n-1}}$$

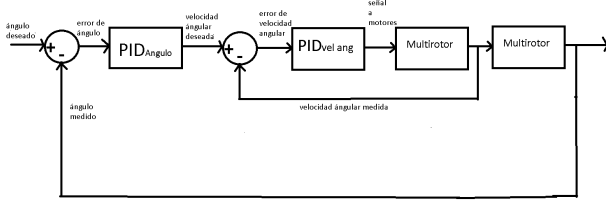
Donde  $I_{anterior}$  se calcula a partir de la parte integral

$$I_{anterior} = k_i e(n)(t_n - t_{n-1}) + I_{anterior}$$

De esta manera se completa el ciclo del PID; ahora resta solamente aplicarlo a cada uno de los ángulos. Para esto nos interesa controlar dos aspectos de cada eje de giro: el ángulo de inclinación en sí y la velocidad angular del mismo eje. Esto se logra utilizando dos controladores **PID** por cada eje mediante etapas como se puede ver en el diagrama con mayor facilidad:

Estos dos **PID** anidados nos permiten controlar ambas variables al mismo tiempo para cada ángulo de manera individual, tomando la retroalimentación de la velocidad angular directamente del giroscopio, y la retroalimentación del ángulo del filtro complementario que se explicó previamente.

Por último, se deben combinar todas las señales a los motores de forma tal que controlen en conjunto el multirrotor. Esto depende de la configuración en que se encuentren los motores; ya que en algunos, esta señal será positiva y en otros negativa. Para el ejemplo de un multirrotor quadróptero en configuración X las señales serían:



$$MotoresPWM_0 = throttle + PID_{roll} - PID_{pitch} - PID_{yaw}$$

$$MotoresPWM_1 = throttle - PID_{roll} - PID_{pitch} + PID_{yaw}$$

$$MotoresPWM_2 = throttle - PID_{roll} + PID_{pitch} - PID_{yaw}$$

$$MotoresPWM_3 = throttle + PID_{roll} + PID_{pitch} + PID_{yaw}$$

#### D. Sintonización

Con todas estas ecuaciones listas, se puede programar al microcontrolador el código para el algoritmo de control. Lo único que resta es la sintonización del PID. Esto se refiere a encontrar los valores de las tres constantes  $k_p$ ,  $k_i$ ,  $k_d$ , para cada uno de los controladores. Existe una rama importante de la ingeniería que estudia la manera de encontrar dichas constantes que se conoce como teoría de control. Mediante modelos matemáticos y distintos criterios de estabilización se pueden encontrar los mejores valores para el PID. Sin embargo, existen algoritmos más sencillos que permiten encontrar estas constantes mediante experimentación, utilizando prueba y error. Se eligen primero valores de  $k_i$  y  $k_d$  muy cercanos a cero. Después se incrementa el valor de  $k_p$  hasta que el multirotor comienza a oscilar, entonces se decrementa ligeramente el valor de  $k_p$ . El tipo de respuesta que se busca es críticamente amortiguado, que es el más rápido. Si oscila puede estar sobre amortiguado, en cuyo caso  $k_p$  debe reducirse, o subamortiguado,  $k_p$  debe incrementarse.

Después, se incrementan gradualmente los valores de  $k_i$  y  $k_d$  de acuerdo a la respuesta que se obtenga y que se quiera lograr. La constante  $k_i$  permite eliminar el error en estado estable, mientras que  $k_d$  permite acelerar la velocidad a la que el controlador responde. Esta forma de encontrar las constantes si bien requiere de mayor tiempo y experiencia, es la forma más sencilla de hacerlo.

De esta forma, es posible ahora enviar un ángulo deseado al microcontrolador, y este se encargará de que el multirotor se mantenga en ese ángulo mientras nosotros lo deseemos, y así moverse en cualquier dirección, o mantenerse estático. Para controlar la posición de forma automática se puede incluir otro **PID** con retroalimentación de GPS y barómetro que tenga como salida el ángulo de inclinación del multirotor.

### III. CONCLUSIÓN

El resultado de este algoritmo puede verse en el programa de Arduino.