

Control de posición de un balancín con DSPIC y módulo de medición inercial.

F. Javier García, Eduardo J. Moya y Pablo Martín, *Miembros de ITAP - Universidad de Valladolid, Paseo del Cauce 59, 47011 Valladolid, España (e-mail: {javgar, edumoy}@eis.uva.es).*

Abstract— En este trabajo se muestran detalles del equipo materializado por el Departamento de Ingeniería de Sistemas y Automática (ISA) de la Escuela de Ingenierías Industriales (EII) de la Universidad de Valladolid (UVA) para la realización de prácticas de laboratorio de la asignatura Fundamentos de Automática presente en los Grados de Ingeniería (conforme a las directrices Bolonia y adaptada al EEES), de las titulaciones impartidas en la EII de la UVA para cientos de alumnos.

En este artículo se muestra el trabajo desarrollado el cual consiste en el control de un balancín con un grado de libertad utilizando 2 motores con hélice en sus extremos, una unidad de mediación inercial o IMU, compuesta por un acelerómetro y un giróscopo, y un Procesador Digital de Señales. Se diseñó un algoritmo de control PID, si bien hubo que filtrar las señales previamente utilizando un filtro de Kalman. Con el fin de facilitar el uso del sistema por los futuros usuarios, se realizó un entorno gráfico utilizando VISUAL BASIC.

Se diseñó y montó el prototipo inicial y en base al conocimiento adquirido, problemas surgidos y solventados, se procedió a la fabricación, en las propias dependencias del departamento, de un total de cuatro equipos.

Index Terms— Balancín, Unidad de medición inercial, Procesador Digital de Señales, filtro de Kalman, Ziegler-Nichols.

I. INTRODUCCIÓN

El estudio la automatización y los sistemas de control es una de las ramas con más importancia en la ingeniería. El estudio de esta disciplina tiene una gran relevancia en el mundo de la docencia electrónica y automática. Este trabajo quiere mejorar y facilitar la comprensión de estos métodos de regulación de sistemas a los alumnos.

Para ello se ha utilizado la automatización de un sistema mecánico. El modelo utilizado es un balancín formado por una barra con dos motores en sus extremos, los cuales crean una fuerza perpendicular a dicha barra al hacer girar una hélice en su extremo. Como variable controlada se tendrá el ángulo girado por el balancín.

El objetivo principal que se pretende es proporcionar a los alumnos una herramienta para entender un sistema de control PID en toda su dimensión, desde el conocimiento de la planta, el diseño matemático o empírico y la programación del control en un microprocesador.

Como objetivos secundarios nos encontramos:

- Proporcionar a los alumnos un sistema de aprendizaje

para entender un sistema de control PID.

- Graficar las distintas señales de entrada, salida e intermedias de un sistema controlado.
- Facilitar las distintas modificaciones de los parámetros de control del sistema, observando las repercusiones de estos cambios sobre el mismo.
- Identificación y obtención de los distintos parámetros de un PID mediante distintos métodos como Ziegler – Nichols, MISAE, MITAE y MISE.
- Entender el funcionamiento de un microcontrolador dsPIC en especial de la serie dsPIC 30.
- Obtener un modelo matemático-físico para poder parametrizar el sistema de manera óptima.

Tras esta breve introducción, el artículo se estructura de la siguiente manera: en la segunda sección se realiza una breve descripción de la planta es decir del sistema mecánico, en la tercera sección se analiza el modelo matemático del sistema, a continuación en la cuarta sección se describe todo el hardware utilizado para el desarrollo de la aplicación. En la sección quinta se describe el filtrado realizado para el correcto funcionamiento de la regulación del sistema, para luego en la sección sexta proceder a realizar el diseño del controlador PID según varios métodos de ajuste. Luego en las secciones seis y siete se muestra el entorno gráfico en el cual se ha desarrollado la aplicación y como se ha desarrollado la propia aplicación. Por último se presentan los resultados y conclusiones obtenidas así como las referencias utilizadas.

II. DESCRIPCIÓN DE LA PLANTA

La planta es un sistema mecánico conformado por un balancín compuesto por una barra a la que se le ha añadido a cada extremo un motor en cuyo eje vertical de movimiento tiene una hélice, la cual producirá un empuje perpendicular a la barra del sistema.

El eje de rotación central del balancín está conformado por un rodamiento, cuya parte externa está dentro de una carcasa, y ésta a su vez está atornillada a una barra que actúa de soporte vertical anclada a la bancada del sistema, mientras que la parte interior del rodamiento está conectado de forma solidaria a la barra del balancín.

La parte móvil del rodamiento se conecta a su vez a un sensor formado por un giróscopo y un acelerómetro.

La bancada está realizada con una placa de acero de cinco milímetros de espesor y está atornillada a una tabla de

aglomerado de madera de 310 x 110 x 20 mm que a su vez soporta una caja en la cual se encuentra la circuitería necesaria para el correcto funcionamiento del sistema, es decir, dos reguladores de voltaje y la placa de circuito impreso, en adelante llamada PCB.



Fig. 1. Sistema físico a controlar formado por la barra giratoria con 2 motores con hélice en sus extremos.

Solidario a la barra se ha colocado un transportador de ángulos que nos permitirá comprobar la posición angular de la barra.

A. Funcionamiento básico

El funcionamiento del sistema está basado en las reacciones de las fuerzas de empuje perpendicular a la barra realizadas por los dos motores. Con la ayuda del transportador de ángulos graficado podemos observar el ángulo de la barra respecto al suelo.

Mediante la modificación del voltaje suministrado a los motores es posible controlar las revoluciones de las hélices de éstos, lo cual repercute en el empuje efectuado a cada extremo del balancín, haciendo que el ángulo del balancín se modifique utilizando el rodamiento como fulcro central del mismo. Éste ángulo a su vez es leído una unidad de mediación inercial (IMU) mediante el método de transmisión de datos I2C, el cual nos indica el giro y la aceleración lineal del sistema. Los datos son recibidos, leídos y procesados por un microprocesador dsPIC. Éste dispositivo tendrá que filtrar los datos recibidos para poder inferir la medida del ángulo.

Los posibles filtros propuestos para la inferencia del ángulo son los filtros tipo Kalman y tipo complementario. Éstos se usan para mejorar las señales recibidas, eliminando así las vibraciones y perturbaciones de la barra, producidas por el motor y las hélices en su movimiento, así como el funcionamiento no lineal del rodamiento del fulcro del balancín.

Tras el filtrado de los datos, el microprocesador realizará un

control mediante un sistema de control PID, y tras ajustes sucesivos, llegará al ángulo final del balancín.

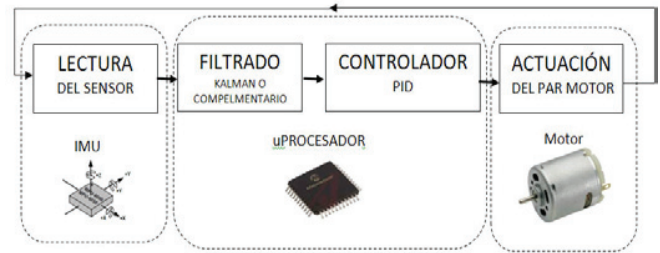


Fig. 2. Diagrama de Bloques del Sistema.

III. MODELO MATEMÁTICO DEL SISTEMA

Para el buen estudio del sistema se necesita un estudio matemático correcto, para ello se estudiará el sistema a tratar, en este caso el balancín, desde un punto de vista de la física teórica.

El estudio se llevará a cabo mediante el estudio de las fuerzas actuantes en el sistema, según la segunda Ley de Newton y su extensión la Ley del equilibrio mecánico, la cual dicta “Una partícula o un sólido rígido está en equilibrio de traslación cuando: la suma de todas las fuerzas que actúan sobre el cuerpo es cero”. En el caso de giros en el espacio, las fuerzas se convierten en momentos, resultando que la suma de momentos es igual a 0.

$$\sum \vec{M}_i = 0 \quad (1)$$

Como nuestro sistema está en equilibrio de traslación, podemos realizar el sumatorio de fuerzas.

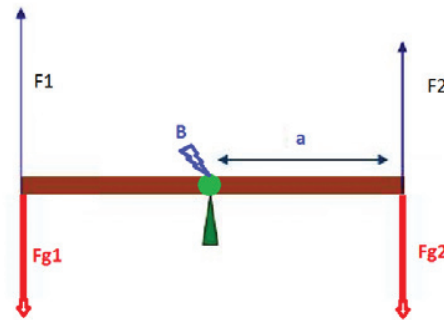


Fig. 3. Diagrama de fuerzas sobre el dispositivo

Eliminando las fuerzas de la gravedad, ya que éstas se contrarrestan, el sistema quedará:

$$\frac{B}{a} \frac{d\theta}{dt} + \frac{i}{a} \frac{d^2\theta}{dt^2} = F_1 - F_2 \quad (2)$$

Tomando Transformada de Laplace suponiendo condiciones iniciales nulas se obtiene la función de transferencia:

$$\frac{\theta(s)}{F_1 + F_2} = \frac{1}{\frac{B}{a}s + \frac{i}{a}s^2} \quad (3)$$

donde: θ es el ángulo
 α es la distancia
 i es el momento de inercia

Y que vemos tiene un polo en el origen, por lo que el sistema en lazo abierto es inestable.

IV. HARDWARE UTILIZADO

En este trabajo se han usado una serie de dispositivos para su correcto funcionamiento y control:

A. Unidad de medición inercial o IMU

Es un dispositivo electrónico funciona detectando la actual tasa de aceleración usando un acelerómetro, y obtiene los cambios en los atributos rotacionales tales como cabeceo, alabeo, y guiñada, usando giróscopos..

Estos dispositivos actualmente están siendo muy estudiados y mejorados ya que son implantados en todo tipo de robots para mejorar su capacidad de movimiento e interacción con el medio, muy en especial para sistemas robóticos aéreos como en quadricópteros y drones.

Una IMU funciona detectando la actual tasa de aceleración usando un acelerómetro, y detecta los cambios en los atributos rotacionales tales como cabeceo, alabeo y guiñada usando giróscopos.

El IMU seleccionado es el MPU-6050 [1]. Este dispositivo forma parte de los llamados sistemas electromecánicos (MEMS), es decir, es un minúsculo sistema mecánico que mediante la microelectrónica sirve para instrumentar valores físicos. Se comunica mediante el protocolo I2C con el microprocesador.

El dispositivo ha sido construido por Inversense y proporciona 6 ejes (3 giros y 3 aceleraciones), los cuales tendremos que procesar de forma matemática para obtener unos datos válidos.

B. Bus de comunicaciones

Para la comunicación con el microprocesador se ha utilizado un bus de comunicaciones en serie con el protocolo I2C. Está siendo ampliamente utilizado en la industria, principalmente para comunicar microcontroladores con sus periféricos en sistemas integrados (Embedded Systems) y, de manera más general, es usado para comunicar circuitos integrados entre sí, los cuales normalmente están en un mismo circuito impreso. Mediante este protocolo se comunica la unidad de medición inercial con el microprocesador. El nombre de este protocolo proviene de Inter-Integrated Circuit (Inter-Circuitos Integrados). La velocidad de transmisión de datos es de 400 kbit/s para el uso con el IMU, siendo ésta la máxima velocidad aceptada. Es comunicación de tipo half dúplex. Comunicación bidireccional por la misma línea pero no simultáneamente.

La principal característica de I2C es que utiliza dos líneas para transmitir la información: una para los datos y otra para la señal de reloj. También es necesaria una tercera línea, pero

ésta sólo es la referencia (masa). Como suelen comunicarse circuitos de una misma placa que comparten una misma masa, esta tercera línea no suele ser necesaria.

Estas líneas distintas se llaman:

-- SDA: datos

-- SCL: reloj

-- GND: tierra

Los datos en este protocolo están estipulados y tienen que seguir una serie estricta para el envío de los mismos.

El dsPIC empleado dispone de un módulo de I2C de alta velocidad integrado siendo así muy simple la configuración.

Las líneas de datos y de reloj están en colector común y para el caso de dsPIC se recomienda colocar unas resistencias de pull-up de 2.2k Ω para que los distintos dispositivos puedan modificar los voltajes, para crear el protocolo.

C. Microprocesador

Como microprocesador se usó un **procesador digital de señales o DSP**, que posee un conjunto de instrucciones, un hardware y un software optimizados para aplicaciones que requieran operaciones numéricas a muy alta velocidad. Debido a esto, es especialmente útil para el procesamiento y representación de señales analógicas a tiempo real.

En este caso, se usó un microprocesador de la compañía Microchips y de la serie DsPIC30, concretamente el DsPIC30f4013 [2], el cual dispone de todas características necesarias para la realización del proyecto.

Este procesador es relativamente nuevo, trabaja con tecnología 16 bits, por lo que facilita su uso en el procesamiento de señales, debido a su rapidez y capacidad para almacenar datos. Puede realizar todas las operaciones necesarias "por ciclo" en un tiempo mínimo de 1ms, lo cual ayuda mucho a la estabilidad del sistema. Presenta 2 pines PWM, 2 módulos de comunicación UART, 1 módulo de comunicación I2C y trabaja a 20 MIPS de velocidad.

Toda la implementación software del sistema se realizó mediante interrupciones-temporizadas, por lo que el sistema cumple con todos los tiempos de ciclo de forma muy exacta.

D. Circuito impreso

El circuito **impreso PCB** se realizó de manera no industrial, con el software PROTEUS, que compila un programa de diseño y simulación electrónica, desarrollado por Labcenter Electronics.

El circuito incluye un microprocesador Dspic, un circuito de acoplamiento para el mismo, una serie de LED de indicación-señalización y una etapa de potencia en un mismo PCB.

La etapa de potencia se realizó mediante unos transistores BD139, los cuales soportan hasta 2 A de intensidad y 30 V de tensión. A la base de estos transistores se conectan los PWM del microcontrolador, con lo cual se puede controlar linealmente la potencia de los motores de forma individual.

También se dispone de dos interruptores conectados a las patillas de interrupción externa del DsPIC, haciendo el procesamiento de estos instantáneo, pudiendo parar los motores en caso de existir un error o problema.

La placa se “ruteó” manualmente, siguiendo criterios de diseño de compatibilidad electromagnética, con el fin de que la parte de control no se vea afectada por la parte de potencia.

V. FILTRADO

Para el correcto funcionamiento de la regulación del sistema, y debido a que nuestro sensor es un acelerómetro-giróscopo, antes del procesamiento de la señal por parte del controlador necesitaremos **un filtro**. Éste tiene como misión corregir los valores erróneos dados por el sensor y fusionar todas las mediadas en una mediada estimada más exacta que las anteriores.

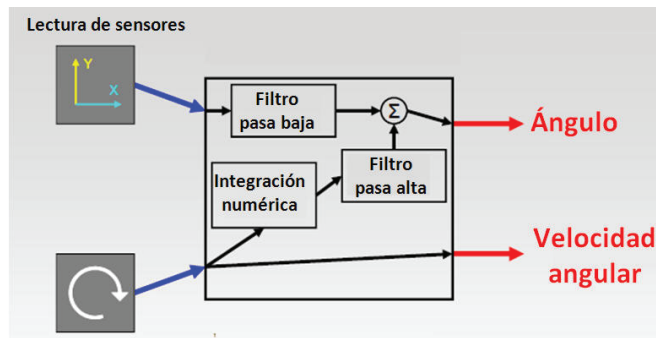


Fig. 5 Esquema gráfico del filtro complementario.

El problema que se observa es que los datos del acelerómetro son fiables sólo a largo plazo, mientras que los datos de giróscopo son fiables sólo a corto plazo, por ello se utilizarán dos tipos de filtros para corregir esto. El primero es un **filtro complementario**, el cual está basado en una ponderación y fusión porcentual de los dos sensores (acelerómetro y giróscopo), un filtro pasa-baja y una integración numérica para obtener resultado ponderado.

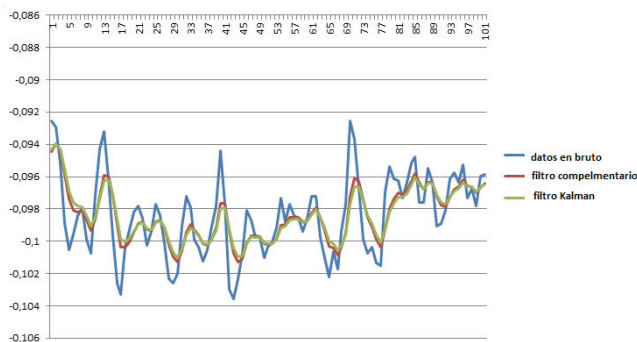


Fig. 6 Gráfica comparativa de los distintos filtros utilizados.

El segundo filtro utilizado es un **filtro Kalman** [3], siendo este un algoritmo que utiliza una serie de mediciones observadas a través del tiempo, en este caso un acelerómetro y un giróscopo, para estimar el ángulo girado de un sensor. Este filtro estima el estado del sistema, con los datos de los estados actuales y los anteriores, que tienden a ser más precisos que los de las medidas instantáneas por sí solas. El filtro de Kalman funciona produciendo una estimación

estadísticamente óptima del estado del sistema basándose en las medidas adquiridas por el acelerómetro y por el giróscopo. Para hacer esto, no solo se necesita saber el ruido que se produce a la entrada al filtro (llamado el ruido de medición), sino también el ruido del propio sistema (llamado el ruido del proceso). Para ello estos ruidos tienen que ser una distribución Gaussiana con media igual a cero. Estos ruidos, al ser aleatorios, poseen esta característica.

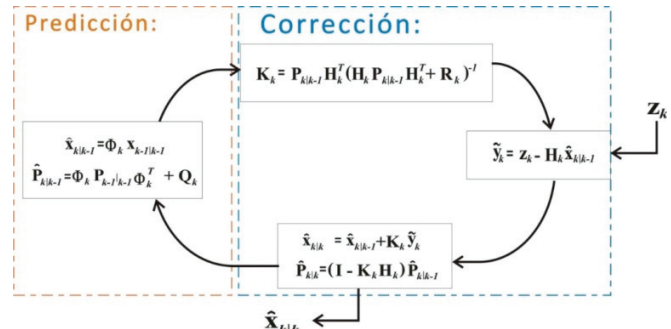


Fig. 7 Esquema de funcionamiento del filtro de Kalman.

VI. DISEÑO DEL CONTROLADOR

El sistema a controlar es claramente un sistema discretizado, ya que se realiza con un microcontrolador y esto hace que la lectura de los sensores se desarrolle en periodos de tiempo establecido, por lo tanto se necesita muestrear la señal para que sea similar a la continua, para ello se utilizarán distintos métodos, llamados métodos de discretización.

Los métodos más importantes son tres:

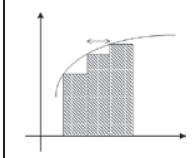
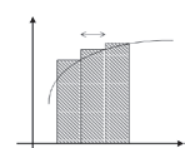
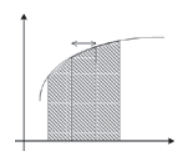
Euler hacia delante	Euler hacia atrás	Trapezoidal
		
$\int_0^t e(\tau) d\tau \approx \sum_{i=0}^{k-1} e(i)T = \sum_{i=0}^{k-1} T e_i$	$\int_0^t e(\tau) d\tau \approx \sum_{i=1}^k e(i)T = \sum_{i=1}^k T e_i$	$\int_0^t e(\tau) d\tau \approx \sum_{i=1}^k T \frac{e_i + e_{i-1}}{2}$

Fig. 8 Método de aproximación discreta.

En nuestro caso, se realizó una aproximación de Euler hacia atrás debido a que nuestro tiempo de muestreo puede ser tan bajo de manera que no se nota de manera real el tipo de método utilizado a efectos prácticos, y aunque se implementaron otros, no se notó mejora. Además este método tiene la virtud de no cargar el microprocesador con operaciones, lo cual mejora los tiempo de ciclado ajustándose más a un sistema continuo y siendo así más fácil de estabilizar el sistema.

Para el algoritmo de control se ha utilizado un **PID** [4]-[7]. Estos controladores han demostrado ser robustos y extremadamente beneficiosos (costo, mantenimiento, soporte) en el control de muchas aplicaciones de importancia en la industria, por lo cual se ha utilizado éste con fines didácticos. El controlador calcula la diferencia entre la señal en momento actual y la señal de punto de consigna, obteniendo así la señal

de error. Esta señal es amplificada por ganancia para realizar una salida proporcional. También se compara la señal de error actual con la señal de error anterior para así crear un término derivativo en la salida, y acumulando el error se obtiene la parte integral. La suma de estas 3 partes conforma una señal que controla nuestra planta.

Para la **configuración de los parámetros del PID** este dispositivo tiene varios métodos de ajuste. Como cualquier PID, el primer método es el método matemático. Usando distintas reglas o criterios de sintonía. Las tratadas en este proyecto son Ziegler-Nichols en lazo abierto y cerrado, MIAE, MISAE, MITAE estas últimas son métodos basado en utilizar el error dinámico como criterio de diseño. Estos métodos son explicados en el proyecto para que sea posible una buena parametrización de las constantes del el PID.

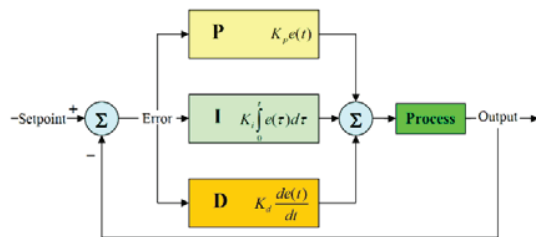


Fig. 9 Esquema del controlador PID.

Otro método tratado a fondo en el proyecto es la sintonización manual de los parámetros. Éste explica cómo sin ninguna tabla y mediante el método empírico, se suelen ajustar los sistemas de control siempre que no sean auto-sintonizados. Además este método se tiene que usar obligatoriamente para un ajuste fino en cualquier método de control PID.

```
//-----calculate P component
float PTerm = ErrorP * _kp;

//-----calculate I component
ErrorI = ErrorI + ErrorP * _ki * Maestreo;

//-----calculate D component
float DTerm = ErrorDT * _kd / Tmuestreo;
```

Fig. 10 Esquema de programación del controlador PID en C.

En el PID se aplicó además un algoritmo Anti Wind-up. El Wind-up es un efecto que produce un retardo en la corrección cuando el término integral excede los límites superior o inferior de la señal de control, es decir cuando se produce una saturación en la salida del controlador. Los sistemas anti wind-up, o de saturación del término integral impiden este fenómeno.

A. PWM

Para proporcionar potencia a los motores se utiliza PWM que es el acrónimo de Modulación por Anchura de Pulso. La modulación por ancho de pulsos de una es una técnica en la que se modifica el ciclo de trabajo de una señal para controlar la cantidad de energía que se envía a una carga.

El ciclo de trabajo de una señal periódica es el ancho relativo de su parte positiva en relación con el período. Expresado matemáticamente:

$$D = \frac{\tau}{T} \quad (4)$$

donde: D: es el ciclo de trabajo.

τ : es el tiempo en que la función es positiva (ancho del pulso).

T: es el período de la función.

Este sistema funciona bien con los motores ya que un motor continuo es gobernado por definición por el valor eficaz del voltaje, definiéndose éste, como el valor de la tensión continua que produce los mismos efectos calóricos que su equivalente de alterna. Es decir, que para determinada corriente alterna, su valor eficaz (I_{ef}) será la corriente continua que produzca la misma disipación de potencia (P) en un resistor (R). Matemáticamente, el valor eficaz de una magnitud variable con el tiempo, se define como la raíz cuadrada de la media de los cuadrados de los valores instantáneos alcanzados durante un período.

En nuestro caso el propio dsPIC nos proporciona 4 módulos PWM de los cuales solo utilizaremos dos de ellos para poder controlar dos motores.

Controlando el PWM podemos controlar las vueltas que dan las aspas, haciendo posible cambiar el empuje por parte de los dos motores.

Es importante la frecuencia de la función, ya que si es demasiado rápida, la etapa de potencia no es capaz de disipar la energía usada en la conmutación, pero si es demasiado lenta, los picos de voltaje que el motor produce al comportarse este como una bobina, pueden estropear el motor y la etapa de potencia.

Hemos elegido un periodo de 6KHz aproximadamente, el cual está dentro del rango recomendable para este tipo de motor y de transistor, recomendable entre 1 kHz y 10 kHz.

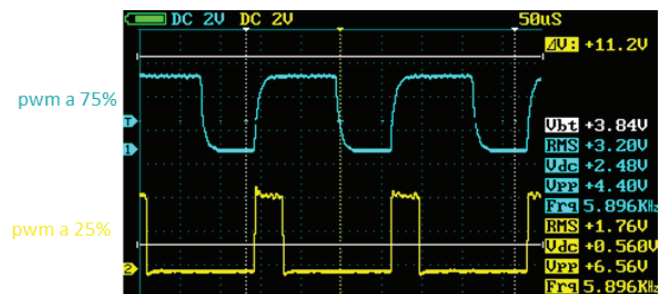


Fig. 11 Gráfica de los 2 PWM.

VII. ENTORNO GRÁFICO DE LA APLICACIÓN.

A fin de facilitar el uso del sistema, así como modificar todo los parámetros, se ha desarrollado una aplicación en VISUAL BASIC [8]. Su objetivo es graficar todas las ondas del sistema (entradas, salida y filtros). Esto será de ayuda para la sintonización del dispositivo, pudiendo así modificar las constantes tanto del PID como de los filtros. También permite la sintonización en lazo abierto y lazo cerrado, siendo este hecho muy práctico a la hora de la sintonización por los distintos métodos de ajuste. Permite guardar tanto las gráficas en formato JPG como los datos recibidos en formato texto, para poder exportar a Excel o cualquier formato de análisis de datos. Además, permite modificar el estado del setpoint del sistema observando la reacción del mismo.



Fig. 12. Pantalla de la aplicación.

Para realizar graficas después de buscar por la red se encontró un programa que tenía todos los requisitos para poder usar como base de nuestro programa final, su nombre era Simplot [9] y tenía estas características:

- Podía graficar desde el puerto serie
- La velocidad de la gráfica muy grande.
- Es software libre
- Tiene muchas posibilidades de modificación y refactorización de código.

Como es software libre tiene una página web donde se puede bajar el código mediante el método mercurial siendo último un tipo de repositorio utilizado en desarrollo informática.

VIII. DESARROLLO DE LA APLICACIÓN

El desarrollo del programa se realizó en C puro, con la aplicación proporcionada por Microchips llamada MPLAB-X.

MPLAB® X IDE es un programa de software que se ejecuta en cualquier ordenador bajo cualquier sistema operativo. Su fin es desarrollar aplicaciones para microcontroladores Microchip, ya que proporciona un único programa donde integra todo lo necesario para desarrollar código para microcontroladores embebidos.

Es programa libre de código cerrado es decir, el propio desarrollador microchips proporciona el programa no así el código fuente del mismo.[10]

En ella, se estructuró el código como programación con fichero código (.c) y ficheros cabecera(.h), para su mejor compilación, pese a que normalmente en programación de microprocesadores se coloca todo en un fichero(.c). Esta opción es más práctica y limpia y muy utilizada en programación orientada a objetos (OOP).

Para programar se utilizó el programador pickit3, con el cual se puede depurar, alimentar y programar. El programa funciona mediante interrupciones y no hay un bucle cíclico.

El programa realiza un bucle infinito sin finalidad aparente, pero como ya se dijo, al crear las interrupciones temporizadas, el programa salta y realiza la interrupción, volviendo al bucle cuando termina con la interrupción. Esto hace que el ciclo de

muestreo sea excesivamente preciso y no se tengan problemas con los tiempos de muestreo.



Fig. 13 Pickit.

El tiempo que tarde la interrupción anterior es indiferente. Lo importante es que la siguiente interrupción siempre salte a un tiempo fijo respecto a la anteriormente establecida por el ciclo de muestreo, porque al iniciar la interrupción lo primero que se hace es configurar la siguiente, para que ésta salte TS (tiempo de muestreo) tiempo después.

Solo puede haber problemas si el tiempo de interrupción es superior al de muestreo, pero esto se ha demostrado imposible experimentalmente.

IX. CONCLUSION

Se ha construido una herramienta que proporciona a los alumnos un sistema de aprendizaje con el fin de que entiendan el método de control PID. En el sistema se pueden ver gráficas de las distintas señales, entrada, salida, intermedia.

Se facilita la modificación de los parámetros de control del sistema, pudiéndose observar sus repercusiones. Se permite la obtención de los parámetros del PID usando diferentes métodos de sintonización como Ziegler-Nichols, MISAE, MITAE y MISE.

Se ha diseñado y construido un prototipo inicial y en base al conocimiento adquirido, problemas surgidos y solventados, se ha procedido a la fabricación, en las propias dependencias del departamento, de un total de cuatro equipos.

REFERENCIAS

- [1] TR4NSDUC7OR, Tutorial de Arduino y MPU 6050, <http://robologs.net/2014/10/15/tutorial-de-arduino-y-mpu-6050/>
- [2] Microchip dspic30f4013 datasheet, <http://ww1.microchip.com/downloads/en/devicedoc/70138c.pdf>, 2004.
- [3] K. Lauszus, "A practical approach to Kalman filter and how to implement it". <http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-filter-and-how-to-implement-it/>, 2012
- [4] J.L. Mírez Tarrillo, "Control PID", http://jmirezcontrol.files.wordpress.com/2012/01/control_pid_jorge_mirez_2012.pdf, 2012, pp. 123–135.
- [5] B. Beauregard "Improving the Beginner's PID – Derivative Kick", <http://bretbeauregard.com/blog/2011/04/improving-the-beginner%E2%80%99s-pid-derivative-kick/>, 2012.
- [6] F. Morilla, "Controladores PID. Ajuste Empírico," http://www.dia.uned.es/~fmorilla/MaterialDidactico/ajuste_empirico.pdf 2006.
- [7] K. Astrom and T. Hagglund, "PID Controllers: Theory, design and tuning." Instrument Society of America (ISA), 2nd Edition 1994.
- [8] J. Serrano Pérez VISUAL BASIC .NET (Manual Avanzado), 2002, ANAYA
- [9] Brijesh, Project Simplot, <https://code.google.com/p/projectsimplot/> (último acceso 23-02-2015)
- [10] <http://www.microchip.com/pagehandler/en-us/family/mplabx/> (último acceso 23-02-2015)