

**“PLATAFORMA DE CONTROL DE MOTORES DC PARA EXPERIMENTACIÓN EN EL
ÁREA DE CONTROL”**

CAMILO ANDRES JIMENEZ GODOY

NICOLAS RIOS PULIDO

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERIA ELECTRONICA
BOGOTÁ D.C.

2011

**“PLATAFORMA DE CONTROL DE MOTORES DC PARA EXPERIMENTACIÓN EN EL
ÁREA DE CONTROL”**

CAMILO ANDRES JIMENEZ GODOY

NICOLAS RIOS PULIDO

TRABAJO DE GRADO PARA OPTAR POR EL TITULO DE INGENIERO ELECTRONICO

DIRECTOR

ING.FREDY O. RUIZ Ph. D.

PONTIFICIA UNIVERSIDAD JAVERIANA

FACULTAD DE INGENIERIA

CARRERA DE INGENIERIA ELECTRONICA

BOGOTÁ D.C.

2011

PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENIERIA
CARRERA DE INGENIERÍA ELECTRÓNICA

RECTOR MAGNÍFICO: JOAQUIN EMILIO SANCHEZ GARCIA, S.J.

DECANO ACADÉMICO: Ing. FRANCISO JAVIER REBOLLEDO MUÑOZ

DECANO DEL MEDIO UNIVERSITARIO: P. SERGIO BERNAL RESTREPO, S.J.

DIRECTOR DE CARRERA: Ing. JUAN MANUEL CRUZ BOHORQUEZ, M.Ed.

DIRECTOR DEL PROYECTO: Ing. FREDY O. RUIZ Ph.D.

ARTÍCULO 23 DE LA RESOLUCIÓN No. 13 DE JUNIO DE 1946

"La universidad no se hace responsable de los conceptos emitidos por sus alumnos en sus proyectos de grado. Sólo velará porque no se publique nada contrario al dogma y la moral católica y porque los trabajos no contengan ataques o polémicas puramente personales. Antes bien, que se vea en ellos el anhelo de buscar la verdad y la justicia".

Artículo 23 de la Resolución No. 13, del 6 de julio de 1946, por la cual se reglamenta lo concerniente a Tesis y Exámenes de Grado en la Pontificia Universidad Javeriana.

INDICE

1.	INTRODUCCIÓN	6
2.	Descripción de la Plataforma	7
3.	Desarrollo	10
3.1	Sistema de control	10
3.1.1	<i>PID</i>	10
3.1.2	<i>Compensación en Atraso</i>	12
3.2	Hardware	13
3.2.1	<i>Tarjeta</i>	13
3.2.2	<i>Micro controlador</i>	13
3.2.3	<i>Motor</i>	14
3.2.4	<i>Puente H</i>	14
3.2.5	<i>Decodificador Óptico</i>	16
3.3	Software	18
3.3.1	<i>Microcontrolador</i>	18
3.3.1.1	<i>Configuración periféricos</i>	18
3.3.1.2	<i>Rutinas</i>	19
3.3.2	<i>Comunicación Serial</i>	28
3.3.3	<i>Interfaz Gráfica</i>	30
3.3.3.1	<i>Panel Frontal</i>	30
3.3.3.2	<i>Diagrama de Bloques</i>	31
4.	Resultados	35
4.1	Frecuencia de Muestreo	35
4.2	Tiempo de Control	36
4.3	Modelado	37
4.4	Control PID de posición	39
4.5	Control Compensador Posición	42
4.6	Control PID de Velocidad	44
5.	Conclusiones	48
6.	Bibliografía	49

1. INTRODUCCIÓN

Dentro del programa de ingeniería electrónica, en el área de control, actualmente no se cuenta con prácticas de laboratorio para las asignaturas de controles y sistemas dinámicos. Solo se desarrollan simulaciones durante la clase o como trabajo personal, lo cual no le permite al estudiante aproximarse a la experimentación antes de los cursos de énfasis, los cuales solo son cursados por los alumnos que están interesados en continuar con sus estudios en esta área.

El énfasis en control ofrece la asignatura de Laboratorio de control. Aquí las prácticas se desarrollan empleando [1] plataformas, estos módulos carecen de versatilidad y su costo es muy elevado. Dentro de las limitaciones que presenta este modulo se encuentran, que su controlador es de tipo PID con estructura fija y solo permite arquitecturas de este tipo. Junto con lo anterior, solo permite evaluar modelos de primer orden, además de esto su tiempo de muestreo es fijo y presenta dificultades a la hora de exportar datos.

Las asignaturas anteriormente mencionadas tienen una gran acogida en los estudiantes ya que estas son la puerta de entrada hacia el énfasis Control. Es en esta rama de la ingeniería donde cada vez se ha despertado mayor interés debido a su versatilidad y creciente campo laboral en la industria nacional y en sectores como la investigación y desarrollo de nuevas tecnologías. Es por estas razones que cobra gran importancia brindarle al estudiante las herramientas necesarias analizar y aterrizar los conceptos recibidos durante el curso de estas asignaturas.

Las prácticas que se han desarrollado en proyectos de grado anteriores [4], se han basado en equipos de National Instruments y requieren de software para sustentar sus operaciones, es por esto que no se cuenta con suficientes equipos, ya que no son de fácil reproducción y limita la cantidad de personas que pueden desarrollar una experiencia en el laboratorio. Con esta limitación de los grupos no es posible para los profesores guiar las prácticas y aprovechar la importancia de las mismas.

Por las razones anteriormente mencionadas en este trabajo de grado se diseñó una plataforma de bajo costo para desarrollar experimentos de caracterización y control en una plataforma sencilla, guiados por el profesor. Esta plataforma no requiere Software especializado, es de fácil reproducción y permite que durante la ejecución de una clase normal se desarrolle la práctica, todo esto con el ánimo de aprovechar los recursos con los que cuenta la facultad.

El objetivo principal consistió en Diseñar un sistema de control de motor DC para el desarrollo de experiencias de laboratorio en el área de control automático. Para llevar a cabalidad este objetivo se tienen los siguientes objetivos específicos:

- Especificar las capacidades de la plataforma experimental para uso en las asignaturas del área de control.
- Diseñar el sistema de control de velocidad y posición para un motor DC basado en un micro controlador.
- Desarrollar la interfaz gráfica que permita cambiar los tipos de controlador, los parámetros del mismo y visualización de gráficas.
- Desarrollar los manuales de uso para la plataforma.

Los primeros capítulos están dedicados a la descripción y las especificaciones de la plataforma, los parámetros para su correcto funcionamiento y las capacidades de la misma. Como acto seguido se describe el proceso que se llevo a cabo para el desarrollo, como la programación del micro controlador, el protocolo de comunicación y el diseño de la interfaz. Por último se encuentran los resultados del desempeño de la plataforma y los diferentes experimentos que se realizaron.

2. Descripción de la Plataforma

La plataforma de control es un sistema que controla un motor DC a partir de una interfaz gráfica, permitiéndole al usuario realizar diferentes tipos de control, generando sus propias señales de referencia y mostrando los resultados en un entorno gráfico.

El sistema de control se encuentra programado en un micro controlador dsPIC30f2010 y para la interfaz gráfica se uso el entorno de programación grafico Labview.

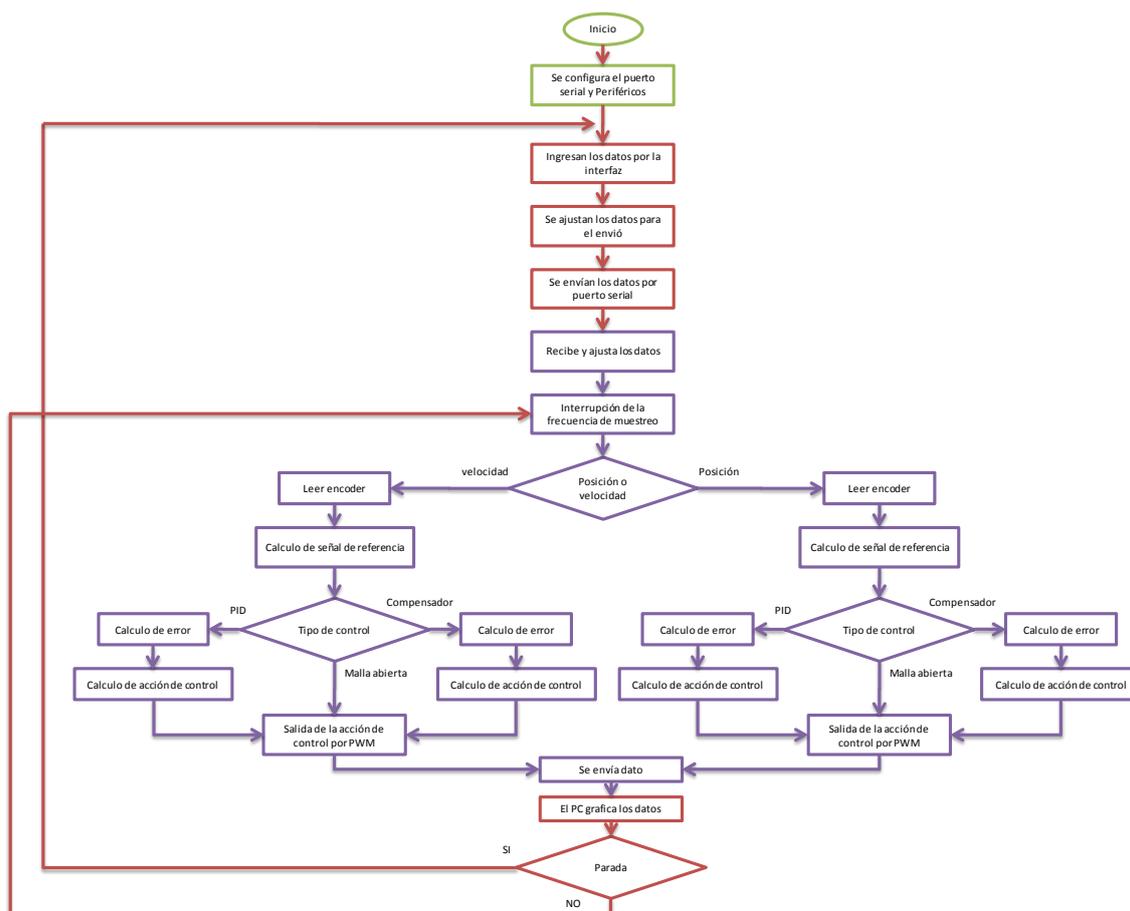


Figura 1 Diagrama de Flujo Plataforma

La figura 1 muestra el funcionamiento del sistema y cómo interactúan la interfaz y el micro controlador durante el proceso. Los bloques que se encuentran en rojo representan la parte del proceso que se realiza en labview, los bloques de color azul corresponden a los procesos que se realizan en el micro controlador y los bloques en verde son procesos ejecutados tanto en el micro controlador como en la interfaz.

Como primera medida se configura el puerto serial para el envío y la recepción de los datos en el computador. Para esto se definió un protocolo de comunicación RS232 con una configuración 8N2 con lo que se van a tener 8 bits de datos y 2 bits de parada, sin paridad. La velocidad de transmisión de los datos es de 57600 bps.

Los datos que se transmiten son ingresados por el usuario en la herramienta, entre estos se incluyó qué tipo de señal de referencia tendrá que ser generada por el micro controlador, la

amplitud y la frecuencia de la misma, así como el tipo de estrategia de control, la acción y los parámetros que serán empleados para el cálculo de la ley de control. Estos datos son ajustados antes de ser enviados con el fin de ser reconocidos por el micro. En la siguiente tabla se pueden observar las variables y su representación.

Dato de entrada	Representación
Tipo de señal	Cuadrada
	Triangulo
	Seno
Frecuencia	1.5 (Hz) - 15 (mHz)
Amplitud	0 - 10 (V)
Frecuencia de muestreo	150 Hz
	75 Hz
	37 Hz
Posición o velocidad	Posición
	Velocidad
Estrategia de control	PID
	Compensación
	Malla abierta
K_d/τ	0 - 1
K_i/α	0 - 1
K_d/K	0 - 1
offset	0 - 10 (V)

Tabla 1 Variables

En el lado del micro controlador fue necesario configurar cada uno de los módulos periféricos que se utilizaron. Entre estos se encuentran, los diferentes timers usados en la programación, el modulador por ancho de pulso, el decodificador óptico y la UART (Universal Asynchronous Receiver Transmitter) la cual es la encargada de la comunicación.

Una vez recibida la palabra se genera la frecuencia de muestreo según fue escogida por el usuario y se inicia el proceso para el cálculo del control. Se definieron tres posibles frecuencias de muestreo 150Hz, 75Hz y 37Hz.

El sistema cuenta con tres posibles señales de referencia que son seleccionadas por el usuario desde la interfaz. Estas son cuadrada, triangulo y seno. La señal cuadrada puede variar entre una amplitud de -10 a 10 voltios pico a pico y una frecuencia desde 15mHz a 1.5Hz. La señal triangulo tiene una amplitud de -8 a 8 voltios pico a pico sin la posibilidad de variar la amplitud ni el offset. La onda seno puede tener una amplitud de 1v,2v, 2.5v, 5v y 10v y offset variable de 0 a 10, las frecuencias de estas señales van de 1.666 Hz a 16 mHz.

Para el sistema de control se toma el dato de posición o velocidad y como se mencionó anteriormente se genera la señal de referencia. Con esto definido se ejecuta la acción de control en malla abierta o en malla cerrada. Como se puede observar en la figura2, el diagrama de bloques muestra como es el proceso para cada caso.

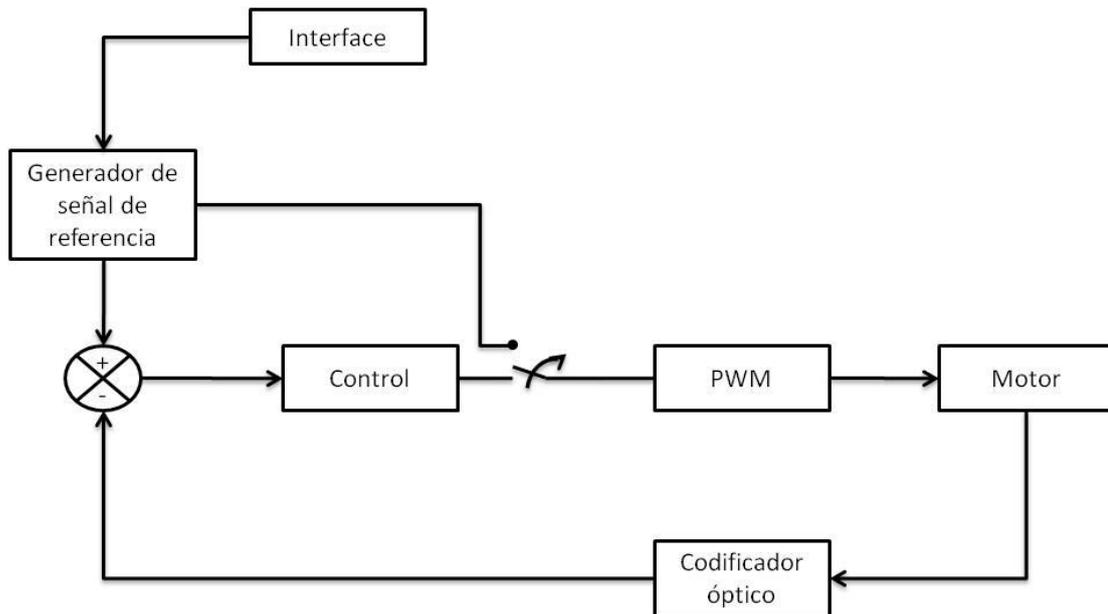


Figura 2 Diagrama de bloques Plataforma

Cuando el sistema ha sido configurado para trabajar en malla abierta el dato que se obtiene del micro controlador es el de la velocidad del motor ante una entrada de voltaje, el cual es suministrado por el codificador óptico directamente. A su vez se reciben en la interfaz los datos de voltaje aplicado al motor y la señal de referencia. Este modo de operación es utilizado para encontrar el modelo del motor.

Cuando el sistema ha sido configurado para trabajar en malla cerrada el dato que se obtiene del micro controlador es el de la posición o el de la velocidad el cual es suministrado por el codificador óptico. A su vez se reciben en la interfaz los datos del Voltaje en el motor y la señal de referencia que se uso durante el cálculo. Para este modo de operación se puede trabajar con dos leyes de control, compensadores en atraso o control PID.

Cada vez que se realiza control, el micro controlador construye la trama que será enviada de regreso al PC, con los datos de posición o velocidad, el voltaje aplicado al motor y la señal de referencia que fue usada para este cálculo.

La interfaz recibe el paquete y luego de un proceso de reconocimiento de la trama, cada dato es gráfico para su posterior análisis. Este proceso se repite una y otra vez, hasta que desde la interfaz, el usuario detiene la marcha del motor. En este punto es posible volver a ingresar los datos con el ánimo de realizar una experiencia nueva.

3. Desarrollo

En el siguiente capítulo se presenta el desarrollo de este proyecto mediante una descripción detallada de cada uno de las tareas que realiza y de cómo fue configurada esta plataforma.

3.1 Sistema de control

En el siguiente capítulo se presentan los procesos de discretización de los sistemas de control, para su posterior implementación en el micro controlador.

3.1.1 PID

El control PID que fue implementado en el micro controlador, se comporta de la siguiente forma:

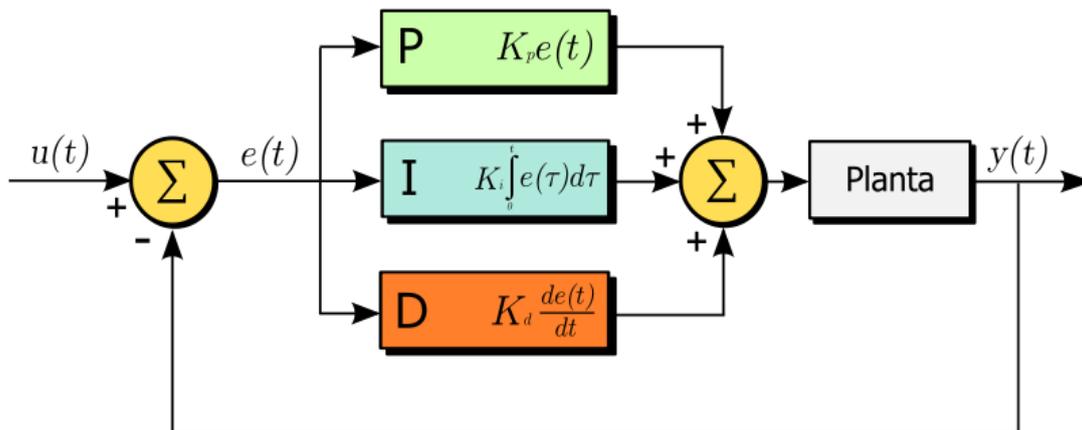


Figura 3 Sistema de control PID

P: La Parte proporcional de un controlador está formada por el producto entre la constante proporcional y la señal de error. Para el parámetro de la constante existe un valor límite para evitar un mal funcionamiento del sistema.

Función de transferencia: $H(S) = K_p$

Parámetros:

K_p = Ganancia proporcional

D: La función de la acción derivativa es mantener el error al mínimo corrigiéndolo proporcionalmente con la misma rapidez que se produce, de esta forma evitar que el error se incremente.

Función de transferencia: $H(S) = K_p + K_d S$

Parámetros:

K_p = Ganancia proporcional

K_d = Ganancia derivativa

I: La parte integral de este controlador cumple con la función de disminuir y eliminar el error en estado estable generado por la parte proporcional, esta parte funciona cuando existe una desviación entre la variable y la señal de referencia, integrando dicha desviación en el tiempo y sumándola a la acción proporcional del sistema.

Función de transferencia: $H(S) = K_p + \frac{K_i}{S} + K_d S$

Parámetros:

K_p = Ganancia proporcional

K_i = Ganancia Integral

K_d = Ganancia derivativa

Para implementar esta acción de control se utilizó la ecuación diferencial del controlador PID y aplicando una aproximación de tipo Euler [8] de la derivada y la integral, obtenemos la siguiente expresión.

Proporcional:

$$U_p(k) = e(k)K_p$$

Integral:

$$U_i(k) = e(k) \frac{K_i}{\Delta t} + U_i(k-1)$$

Derivativa:

$$U_d(k) = \frac{K_d}{\Delta t} (e(k) + e(k-1))$$

PID:

$$U_c(k) = U_p(k) + U_i(k) + U_d(k)$$

Con base en estas ecuaciones se implementó el control digital, no sin antes tener en cuenta el efecto de WIND-UP, para poder evitar este fenómeno, se dispuso de una comparación de la acción de control con los valores máximo y mínimo de la acción integral, estos valores corresponden a los límites del PWM que van desde 0 hasta 1000 conteos.

3.1.2 Compensación en Atraso

La principal función del compensador en atraso es brindar una atenuación en la banda de frecuencias altas con el fin de contribuir con un margen de fase suficiente para el sistema, cabe anotar que la característica de atraso de fase no tiene ningún efecto sobre la compensación en atraso.

$$G_c(s) = K \frac{\tau s + 1}{\tau \alpha s + 1}$$

Para la discretización de este controlador se empleo la transformación bilineal o regla de Tustin [8]

$$s = \frac{2}{\Delta t} \frac{z - 1}{z + 1}$$

$$G_c(z) = K \frac{\tau \frac{2}{\Delta t} \frac{z - 1}{z + 1} + 1}{\tau \alpha \frac{2}{\Delta t} \frac{z - 1}{z + 1} + 1}$$

$$G_c(z) = K \frac{2\tau(z - 1) + \Delta t(z + 1)}{2\tau\alpha(z - 1) + \Delta t(z + 1)}$$

$$G_c(z) = K \frac{2\tau z + \Delta t z - 2\tau + \Delta t}{2\tau\alpha z + \Delta t z - 2\tau\alpha + \Delta t}$$

$$G_c(z) = K \frac{2\tau z + \Delta t z - 2\tau + \Delta t}{2\tau\alpha z + \Delta t z - 2\tau\alpha + \Delta t}$$

$$G_c(z) = K \frac{z(2\tau + \Delta t) + (\Delta t - 2\tau)}{z(2\tau\alpha + \Delta t) + (\Delta t - 2\tau\alpha)}$$

$$G_c(z) = K \frac{(2\tau + \Delta t) + (\Delta t - 2\tau)z^{-1}}{(2\tau\alpha + \Delta t) + (\Delta t - 2\tau\alpha)z^{-1}}$$

$$a = (2\tau + \Delta t)$$

$$b = (\Delta t - 2\tau)$$

$$c = (2\tau\alpha + \Delta t)$$

$$d = (\Delta t - 2\tau\alpha)$$

$$G_c(z) = K \frac{a + bz^{-1}}{c + dz^{-1}}$$

$$\frac{U_c(z)}{e(z)} = K \frac{a + bz^{-1}}{c + dz^{-1}}$$

$$U_c(z)(c + dz^{-1}) = e(z)(Ka + Kbz^{-1})$$

$$U_c(z)c + U_c(z)dz^{-1} = e(z)Ka + e(z)Kbz^{-1}$$

$$U_c(z)c + U_c(z)dz^{-1} = e(z)Ka + e(z)Kbz^{-1}$$

$$U_c(z) = \frac{e(z)Ka + e(z)Kbz^{-1} - U_c(z)dz^{-1}}{c}$$

$$U_c(z) = e(z) \frac{Ka}{c} + e(z)z^{-1} \frac{bK}{c} - U_c(z)z^{-1} \frac{d}{c}$$

$$U_c(k) = e(k) \frac{Ka}{c} + e(k-1) \frac{bK}{c} - U_c(k-1) \frac{d}{c}$$

3.2 Hardware

3.2.1 Tarjeta

Se empleó la tarjeta de desarrollo utilizada en el curso de control digital, en donde se integra al micro controlador con los diferentes periféricos necesarios para el desarrollo de la plataforma. Esta tarjeta cuenta con un micro controlador dsPIC30F2010, un regulador de voltaje LM2594 encargado de la polarización de los diferentes circuitos, una etapa de potencia para el motor configurada en puente H L6203 y un driver compatible para la comunicación RS232 SN65C3221.

3.2.2 Micro controlador

El micro controlador escogido es un DSPIC30f2010. Este es un micro controlador de 28 pines y cuenta con los periféricos que fueron utilizados durante este desarrollo. Su CPU es de 16bits, tiene incluida una memoria RAM con 512 bytes y un Kbyte de EEPROM no volátil. En la figura 4 se encuentra el microcontrolador y el orden de sus pines.

28-Pin SDIP and SOIC

MCLR	1	28	AVDD
EMUD3/AN0/VREF+/CN2/RB0	2	27	AVSS
EMUC3/AN1/VREF-/CN3/RB1	3	26	PWM1L/RE0
AN2/SS1/CN4/RB2	4	25	PWM1H/RE1
AN3/INDX/CN5/RB3	5	24	PWM2L/RE2
AN4/QEA/IC7/CN6/RB4	6	23	PWM2H/RE3
AN5/QEB/IC8/CN7/RB5	7	22	PWM3L/RE4
VSS	8	21	PWM3H/RE5
OSC1/CLKI	9	20	VDD
OSC2/CLKO/RC15	10	19	VSS
EMUD1/SOSCI/T2CK/U1ATX/CN1/RC13	11	18	PGC/EMUCA1RX/SD11/SDA/RF2
EMUC1/SOSCO/T1CK/U1ARX/CN0/RC14	12	17	PGD/EMUDU1TX/SD01/SCL/RF3
VDD	13	16	FLTA/INT0/SCK1/OCFA/RE8
EMUD2/OC2/IC2/INT2/RD1	14	15	EMUC2/OC1/IC1/INT1/RD0

Figura 4 dsPIC30f2010

3.2.3 Motor

La plataforma cuenta con un motor PITTMAN 9233 con escobillas. Su momento de inercia es de $4.2 \times 10^{-6} \text{ Kg/m}^2$. El voltaje máximo que se le aplica es de 10 V y cuenta con una carga adherida al extremo del eje de 39 mm de radio, 8mm de altura y 78 g de peso. En la figura 5 se puede observar el motor que hace parte de la plataforma



Figura 5 Motor Pittman

3.2.4 Puente H

Como se mencionó anteriormente la tarjeta cuenta con un circuito L6203 el cual constituye la etapa de potencia. Este recibe la señal proveniente del PWM y según el ciclo útil de la misma determina el sentido de giro que tendrá el motor. Como se puede observar en la Figura 6 se tienen dos caminos posibles por donde circula la corriente. Una señal de enable representada con un alto, se envía desde el micro controlador para habilitar este dispositivo. En la figura 6 se puede observar el esquemático correspondiente al puente H.

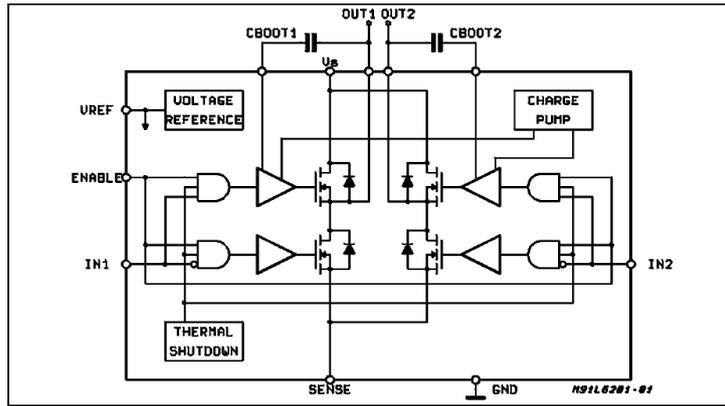


Figura 6 Funcionamiento Puente H

El PWM fue configurado directamente en el micro controlador para trabajar a una frecuencia de 20kHz. Para garantizar esta frecuencia se calcula el número de conteos que corresponden a la mitad del ciclo, los cuales dependen de la frecuencia del micro controlador esta es de 10MHz.

$$PTPER = \frac{F_{cy}}{F_{pwm} * Prescaler}$$

$$PTPER = \frac{10MHz}{20k * 1} = 500$$

Este número de conteos es equivalente al 50% del ciclo útil, como se puede observar en la Figura 7 para un ciclo útil del 100% se tiene un numero de 1000 conteos.

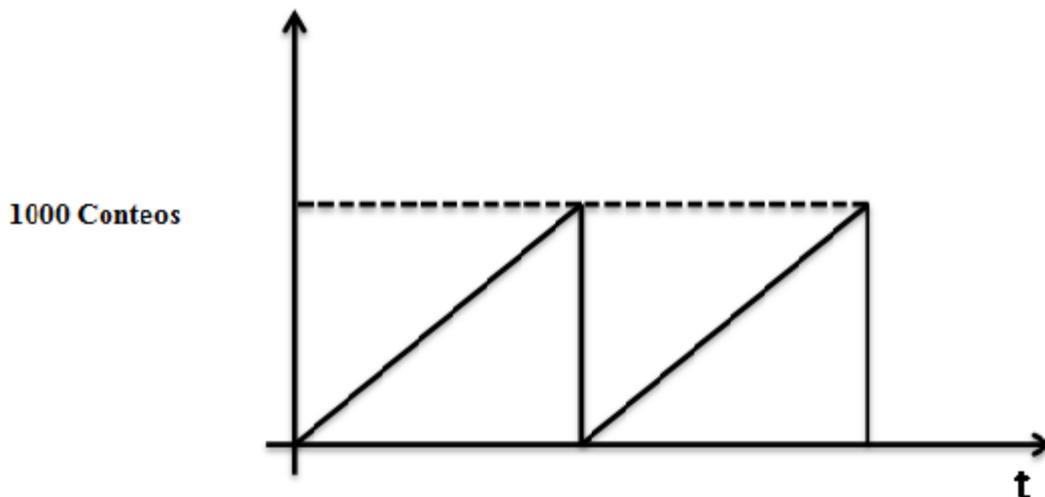


Figura 7 Conteos PWM

3.2.5 Decodificador Óptico

El motor tiene un decodificador Hewlett Packard HEDS, este cuenta con una resolución de 1024 pulsos por vuelta. Esta polarizado con un voltaje de 5V y cuenta con dos canales en cuadratura, a través de los cuales se dirige la lectura del decodificador óptico hacia el micro controlador. En la figura 8 se encuentra el diagrama del encoder en donde se pueden observar los diferentes comparadores, sus canales y como funciona.

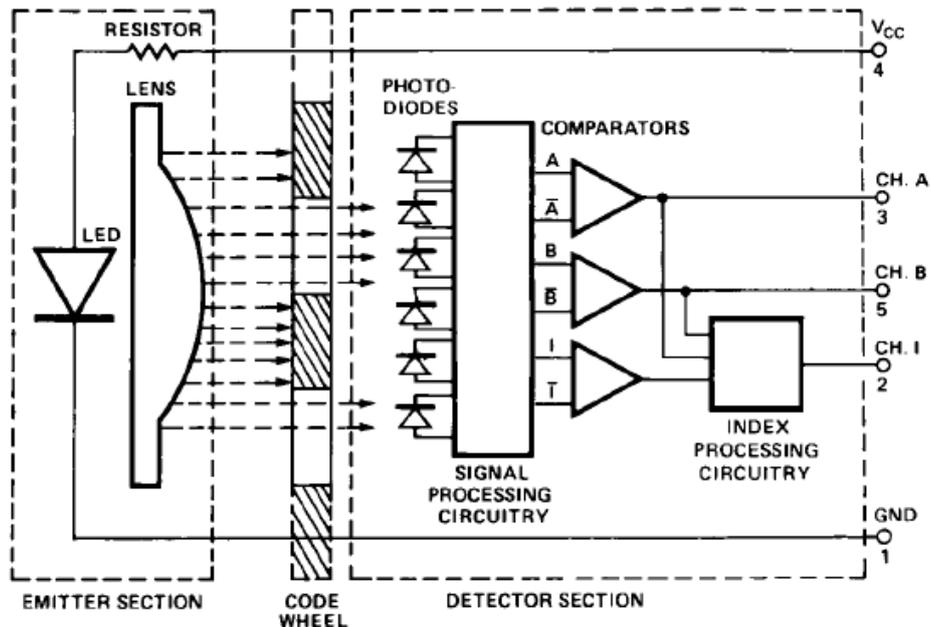


Figura 8 Diagrama de bloques Encoder

Como se mencionó el codificador óptico tiene un máximo de 1024 pulsos por vuelta, cuando se ejecuta un control de posición el rango dinámico máximo será de 500 pulsos o media vuelta, ya que la señal de referencia llega a un valor máximo de 500 conteos. Con la lectura del codificador se calcula la posición del motor. Como se observa en la Figura 9 la posición inicial equivale a 0 pulsos y la máxima a 500.

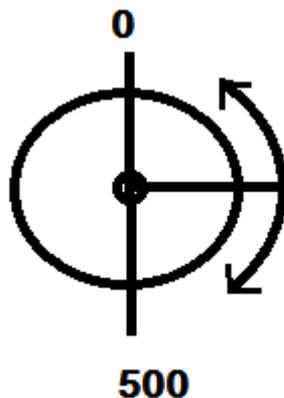


Figura 9 Rango Dinámico Posición

La velocidad del motor está dada por el voltaje que se le suministre, en el caso concreto del sistema que se está tratando la velocidad máxima corresponde al valor máximo de voltaje, cuando al motor se le aplica un voltaje de 10 V la velocidad que alcanza es de 82 rad/s. para capturar el dato de la velocidad se lee el decodificador dos veces con un intervalo de tiempo de 6.6666 ms.

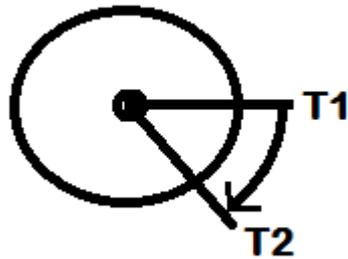


Figura 10 Rango Dinámico Velocidad

$$Velocidad = \frac{\#Pulsos}{T_2 - T_1}$$

3.3 Software

El funcionamiento de la plataforma va a estar regida por su software y la interacción entre el micro controlador y la interfaz gráfica. Aquí se puede encontrar como se realizó el desarrollo del software en el micro controlador y en la interfaz gráfica, su funcionamiento y las herramientas que fueron usadas.

3.3.1 Microcontrolador

Este micro controlador basa su funcionamiento en la ejecución de las rutinas que le fueron programadas en lenguaje C, estas van siendo ejecutadas a lo largo de un ciclo continuo cumpliendo con las tareas que les fueron asignadas. Esta programación se realizó en el software que provee microchip llamado Mplab, junto con esto se utilizó un programador de microchip PICkit3.

3.3.1.1 Configuración periféricos

Para iniciar la programación del micro controlador se configuraron los diferentes periféricos que se utilizaron, estos fueron:

- Timer1
- Timer3
- PWM
- Encoder
- UART
- Interrupciones

Estas configuraciones se realizaron según el Family Reference Manual, este manual explica la operación de los micro controladores de la familia dsPIC30F y brinda la información necesaria acerca de su arquitectura y de sus módulos periféricos.

El timer3 se utilizó para configurar la frecuencia de muestreo. Este timer funciona con el reloj interno, que debe ser configurado con el bit T3CONbits.TCS=0, es con este reloj que el timer se va a sincronizar. El bit T3CONbits.TCKPS=0b00 establece un valor de pre escala con una relación de 1:1 es decir la frecuencia será dividida por un factor de 1. El bit que habilita al timer es el T3CONbits.TON=1, por último se calculó el registro PRx el cual indica el valor máximo de conteo que se debe alcanzar para satisfacer el periodo deseado. Este cálculo se realizó como se observa en la ecuación.

$$Tiempo\ por\ ciclo = \frac{1}{4f_{cy}} = 400ns$$

$$\text{Tiempo de muestreo} = \frac{1}{150} = 6.66\text{ms}$$

$$PR3 = \frac{\text{Tiempo de muestreo}}{\text{Tiempo por ciclo}} = \frac{6.66\text{ms}}{400\text{ns}} = 16666$$

El PWM se configuró en modo free run con el bit PMOD1, la relación de pre escalización es de 1:1 y se habilita la base de tiempos con el bit PTEN. El registro PTPER determina el periodo del mismo.

Para el modulo del codificador óptico se configuró el modo 'x4' para tener una mayor resolución y se tiene a el MAXCNT como el límite de posición, volviendo a iniciar el conteo cuando se alcanza el valor que se encuentra en este registro. Trabaja con reloj interno y un factor de escala de 1:1. Para este modulo, se empleo un filtro digital que rechaza picos de ruido que se presentan normalmente en aplicaciones como esta. Este filtro asegura que las señales de salida no van a a cambiar hasta que no se haya registrado un valor estable durante tres ciclos consecutivos del filtro. El filtro esta activado por el bit QEOUT=1 y esta dividido por un factor de 1.

El modulo UART (Universal Asynchronous Receiver Transmitter), es el modulo dedicado a la comunicación con dispositivos periféricos en este caso con el puerto serial del computador. El bit PDSEL=0b0 indica que se van a tener 8bits de datos, sin paridad y se configuro para 2bits de parada con el bit STSEL=1.

Las interrupciones fueron configuradas para producir el tiempo de muestreo y para la recepción de los datos. Los niveles de prioridad que se seleccionaron fueron 7 para la recepción y 5 para la interrupción del timer, estos niveles van de 1 a 7 con 7 como la máxima prioridad.

3.2.1.2 Rutinas

Main:

La función main es el cuerpo principal del programa encargado de operar el micro controlador, debido a que en esta función se declaran las variables globales, se declaran las sub funciones que se utilizan en todo el programa y se aloja el while principal del programa. Dentro del while principal se ajustan los datos que recibe del PC para poder operar sus rutinas, estos ajustes se realizan con el fin de identificar el dato recibido y si es necesario hacer una conversión que ajuste el parámetro a lo requerido por el micro, además de esto, está encargado de enviar los datos hacia el computador cada vez que se produzca una bandera dada por la interrupción de la frecuencia de muestreo, si el usuario desea cambiar los parámetros esta función es capaz de limpiar los datos de control y cargar nuevamente los parámetros del sistema solo si previamente se recibió el dato para detener el motor. En la *Figura10* se puede observar el diagrama de flujo correspondiente a la rutina.

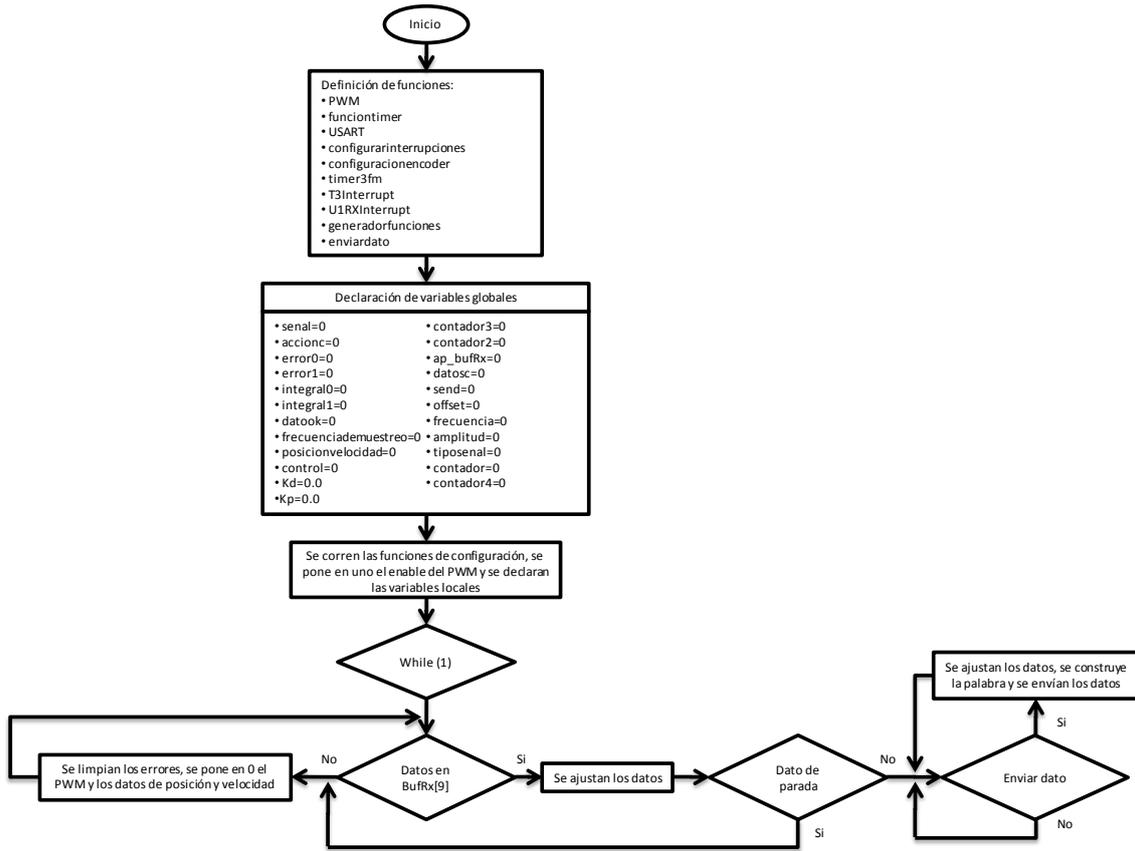


Figura 11 Diagrama de flujo Main

U1RXInterrupt

Esta función está configurada para tener la máxima prioridad de las interrupciones puesto que de esta depende la carga de los datos, esta rutina cuenta con un arreglo de 10 posiciones en el cual se encuentran todos los parámetros y las opciones de control que se disponen en la interfaz. Cada vez que se genera una interrupción la función llena una de las posiciones del arreglo y aumenta en uno un apuntador que recorre el arreglo, cuando el apuntador alcanza el valor de 9 se le asigna un valor de cero para finalizar la carga de datos y dejar listo el sistema para llenar la siguiente palabra que se envíe desde el PC.

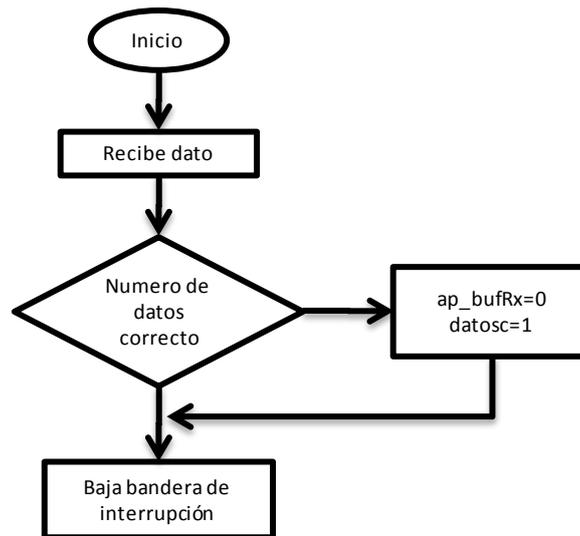


Figura 12 Diagrama de Flujo U1RX1Interrupt

La *figura 12* representa el diagrama de flujo para la interrupción de U1RX, una vez se reciben los datos requeridos por el sistema, estos son cargados.

Cuadrada

La función cuadrada es encargada de generar la señal de referencia con la forma de onda cuadrada mediante un contador y los parámetros de entrada frecuencia, amplitud y offset. La suma de un offset de 500 y un rango de 0 a 1000 con amplitud máxima de 500 es de suma importancia puesto que estos valores coinciden con los valores del registro encargado de operar el PWM. Esta rutina es capaz de variar sus parámetros de amplitud y frecuencia, esto es posible mediante las siguientes operaciones.

$$\text{Ciclo útil} = \text{frecuencia} / 2;$$

$$\text{Alto} = \text{amplitud} + 500 + \text{offset};$$

$$\text{Bajo} = 500 - \text{amplitud} + \text{offset};$$

El contador se encarga de mantener la señal en alto si el conteo es menor al valor del ciclo útil y en alto si es mayor al ciclo útil. Cuando el contador alcanza el valor de la frecuencia vuelve a cero y pone la señal en bajo, para poder iniciar el siguiente ciclo. La *figura 13* representa el diagrama de flujo para esta rutina,

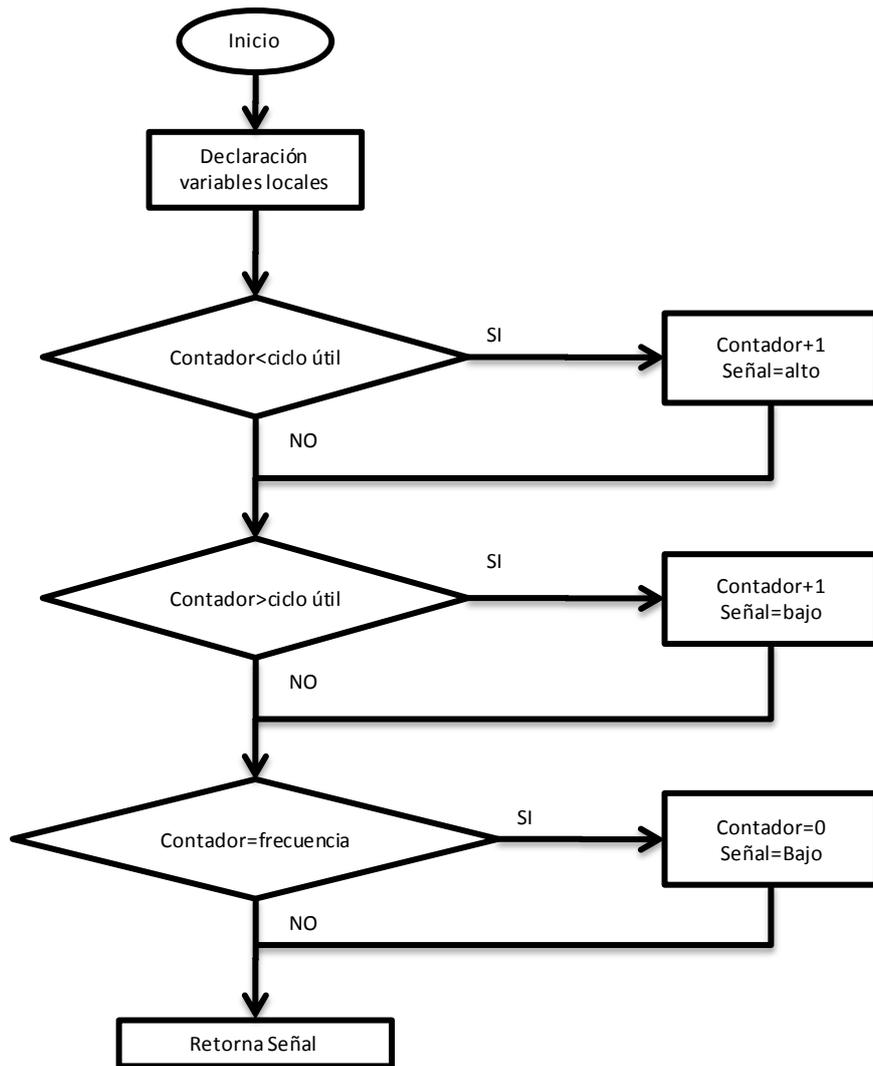


Figura 13 Diagrama de flujo Cuadrada

T3Interrupt

Esta interrupción es la encargada de efectuar las acciones de control y activa si el sistema tiene los datos correctos que provienen del PC y con un tiempo de 6.66 ms, este tiempo corresponde a una frecuencia de 150 Hz, esta rutina también tiene la capacidad de efectuar la acción de control con una frecuencia de 75 Hz y 37 Hz, estos valores corresponden a la frecuencia inicial dividida por 2 y por 4 respectivamente.

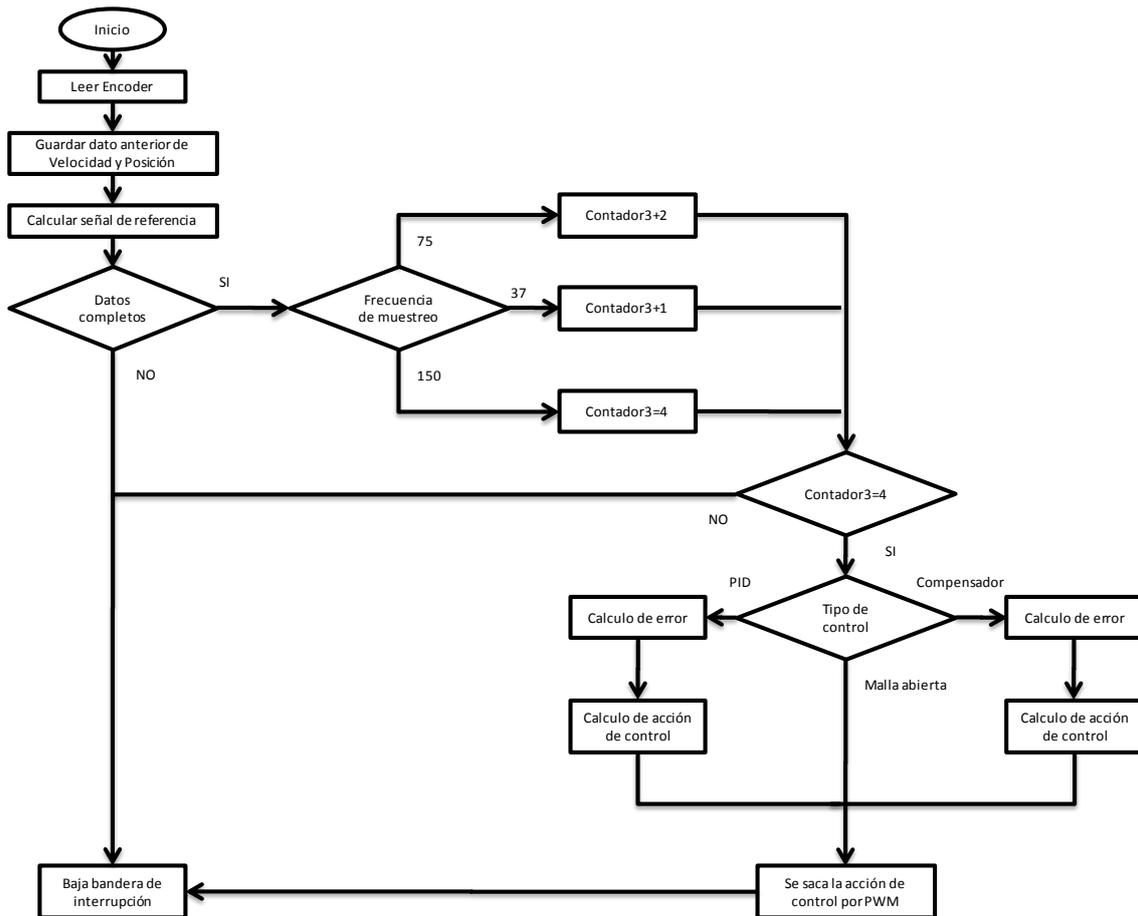


Figura 14 Diagrama de flujo T3Interrupt

Como se observa en la *Figura 14* al inicio de la interrupción se puede encontrar la lectura del encoder y el cálculo de la velocidad del motor con estos datos y dependiendo de la acción de control y el tipo de señal de referencia que haya seleccionado el usuario el sistema calcula el error entre la señal y el estado del motor, una vez efectuado el cálculo de dicho error el sistema realiza la acción de control que corresponda a la escogida por el usuario, posterior a esto el sistema modula este error por PWM con el fin de suministrarle el valor adecuado de voltaje al motor.

Generadorfunciones:

La tarea de esta función es seleccionar la rutina que corresponda a la selección de la señal de referencia que el usuario designo y mandarle los datos de frecuencia, amplitud y offset a la rutina encargada del cálculo de cada señal. En la figura 15 se encuentra el diagrama de flujo para esta función.

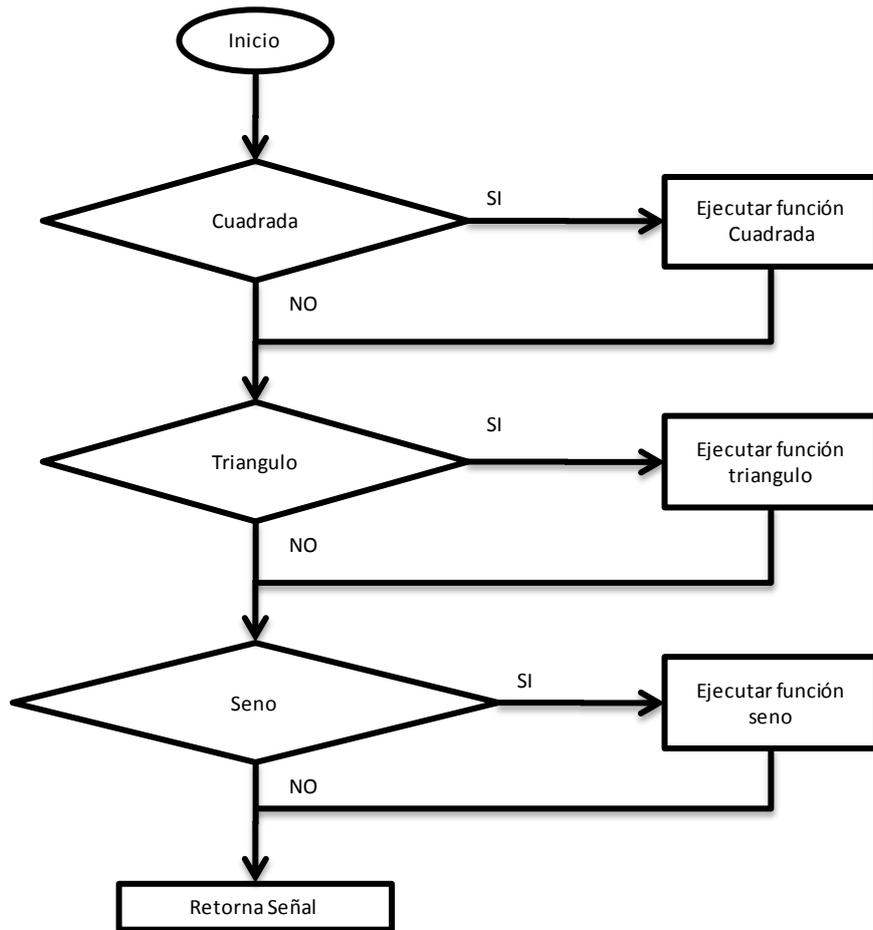


Figura 15 Diagrama de flujo Generadorfunciones

Enviar dato

Como su nombre lo indica la tarea de esta rutina es enviar los datos del micro controlador al computador, para poder realizar esta tarea es necesario cargar los datos que se desean enviar en un arreglo, este cuenta con una longitud de 8 bits. Como los datos que van a ser enviados, voltaje del motor, acción de control y posición o velocidad, son números de 10 bits, es necesarios llenar dos arreglos para poder formar una palabra de 10 bits, para efectuar esta labor, se llenan los bits menos significativos en el segundo arreglo y los bits más significativos en el primer arreglo, también es importante mandar un byte al inicio y al final de la trama, una cabeza y una cola, para esto se disponen en los arreglo en la primera y en la última posición con un valor de cero, para garantizar que los arreglo que contienen los datos no sean cero se añade un valor de 256 a cada dato y se pone una condición para que el arreglo con los bits menos significativos nunca sean iguales a cero. Una vez realizadas estas tareas es posible enviar la trama completa debidamente ordenada la cual cuenta con 8 caracteres de 8 bits (64 bits en total). En la figura 16 se puede observar cómo se desarrolla este proceso.

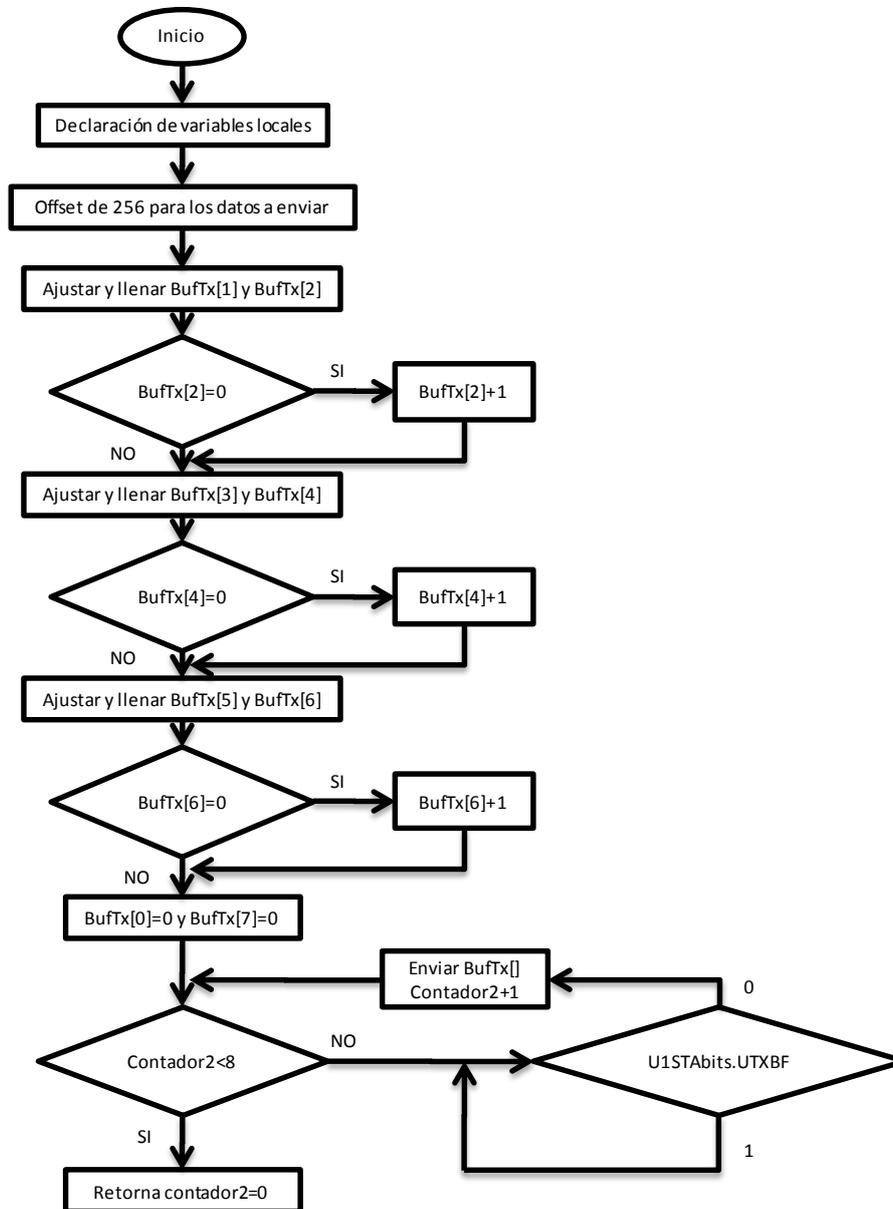


Figura 16 Diagrama de flujo EnviarDato

Seno

El propósito de esta función es generar una señal de referencia con forma de onda seno, para lograr esto es necesario crear una tabla y recorrerla en todos sus puntos, esta tabla fue generada con la ayuda de Matlab y cuenta con 100 caracteres representados en hexadecimal, la frecuencia de esta señal puede ser variada según la velocidad con que se recorra la tabla y la amplitud se puede variar dividiendo el valor de la tabla por un factor. En la figura 17 se encuentra el diagrama de flujo para esta rutina.

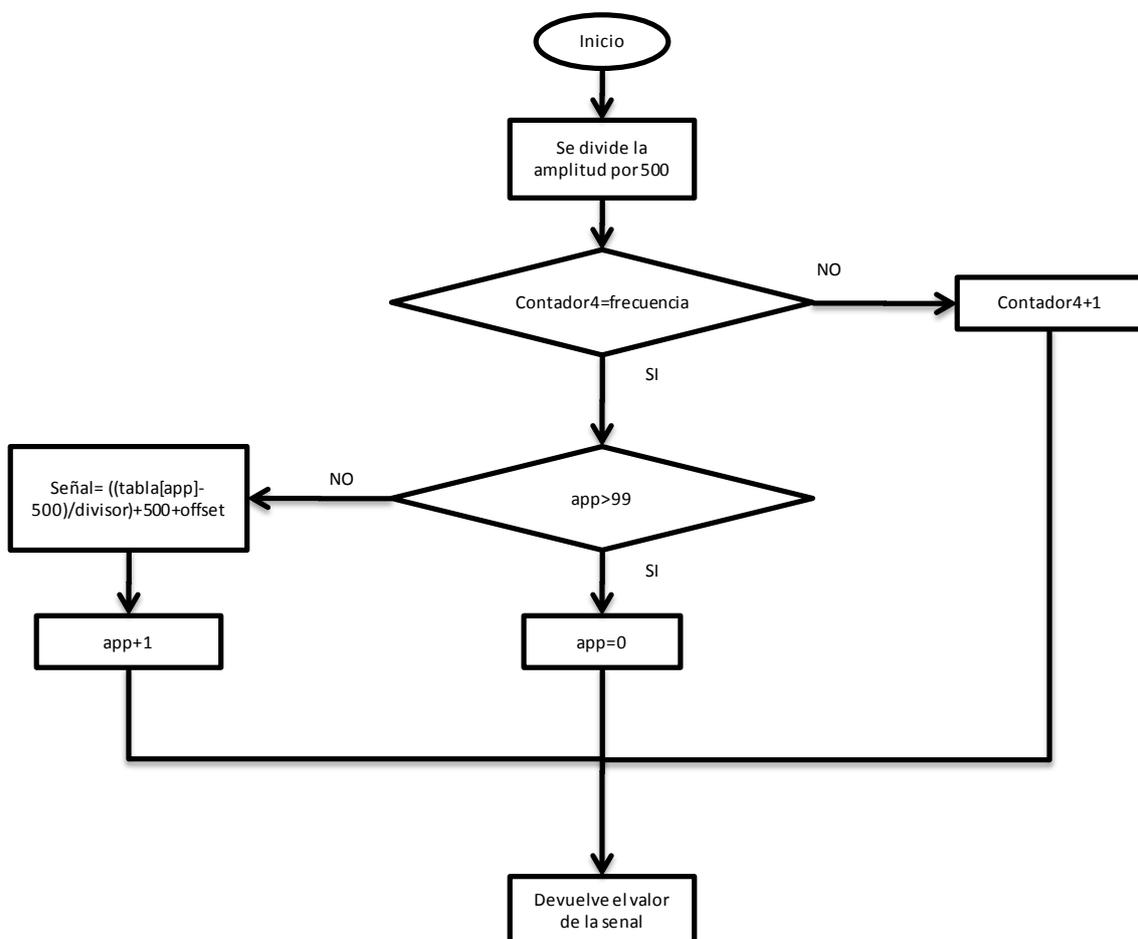


Figura 17 Diagrama de flujo Seno

Triangulo

Esta rutina es la encargada de generar la señal de referencia con forma de onda triangular, para lograr este propósito se utilizó un contador y a medida que este va aumentando se le suma a la señal de referencia 16 cuando el contador llega a la mitad del valor de la frecuencia empieza a restar 16 hasta alcanzar el total de la frecuencia. La figura 18 muestra la rutina encargada de generar la señal triangular.

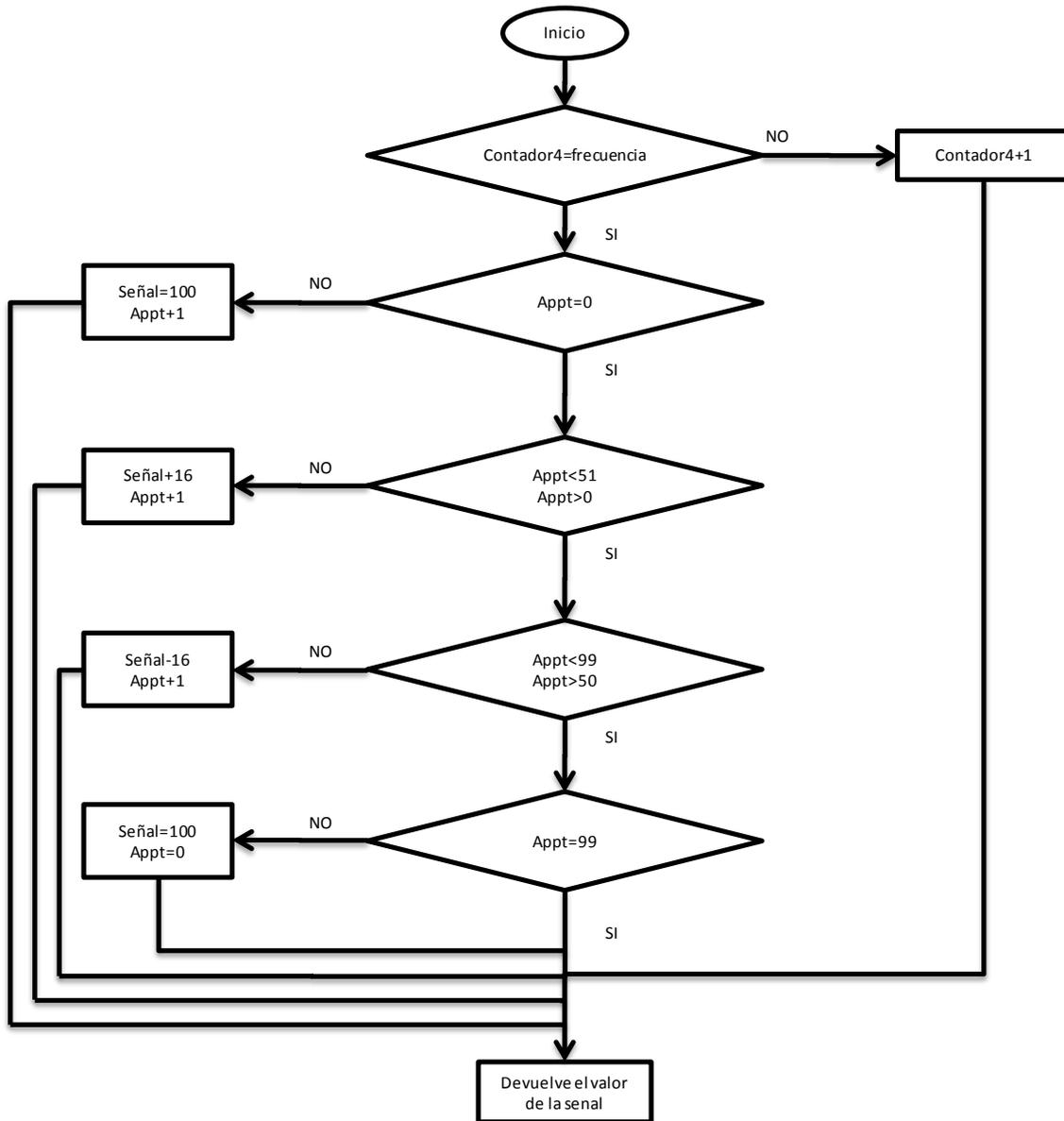


Figura 18 Diagrama de flujo Triangulo

3.3.2 Comunicación Serial

Para la comunicación serial entre el micro controlador y el PC se uso el periférico UART del cual se menciona anteriormente su configuración y la rutina con la cual se envían y se reciben los datos en el micro controlador. En el PC se implemento en Labview una sesión llamada VISA que habilita y configura el puerto según los parámetros deseados en el capítulo siguiente se encuentra explicado en detalle este bloque y su funcionamiento.

Como se describió en el capítulo 2 se implementó un protocolo de comunicación RS232 con 8 bits de datos y 2 de para, sin paridad y una tasa de transferencia de 57600. Para cumplir con estas condiciones fue necesario realizar las siguientes configuraciones.

En la interfaz la configuración se realizó ingresando los valores correspondientes al número de bits, cantidad de bits de parada, paridad y baud rate. En la tabla se puede observar como es armada la trama que contiene las variables configuradas por el usuario y es enviada al micro controlador.

Posición en el BufRx	Dato de entrada	Valores	Ajuste	Valor ajustado	Representación
0	Tipo de señal	1	NA	1	Cuadrada
		2	NA	2	Triangulo
		3	NA	3	Seno
1	Frecuencia	1 - 100	X100	100 - 10000	1.5 (Hz) - 15 (mHz)
2	Amplitud	0 - 100	x5	0 - 500	0 - 10 (V)
3	Frecuencia de muestreo	1	NA	1	150 Hz
		2	NA	2	75 Hz
		3	NA	3	37 Hz
4	Posición o velocidad	1	NA	1	Posición
		2	NA	2	Velocidad
5	Tipo de control	1	NA	1	PID
		2	NA	2	Compensación
		3	NA	3	Malla abierta
6	Kd/τ	0 - 100	÷100	0 - 1	0 - 1
7	Ki/α	0 - 100	÷100	0 - 1	0 - 1
8	Kd/K	0 - 100	÷100	0 - 1	0 - 1
9	offset	0 - 100	x5	0 - 500	0 - 10 (V)

Tabla 2 Trama Variables Interfaz

En el microcontrolador se configuraron los registros que definieron los 8 bits de datos y los dos de parada correspondientes. En la ecuación se puede observar el cálculo que se realizo para configurar la velocidad de transmisión (Baud Rate) a un valor de 57600.

$$UxBRG = \frac{FCY}{16 * Baud Rate} - 1$$

$$UxBRG = \frac{10MHz}{16 * 57600} - 1 = 10$$

$$Baud Rate = \frac{FCY}{16 * (UxBRG + 1)}$$

Con el periodo del registro UxBRG, el cual controla un timer de 16bits se configura el modulo para tener la velocidad deseada

En la figura se puede observar la trama que es regresada al PC desde el micro controlador, esta cuenta con un encabezado y una cola con los cuales se realizo el reconocimiento de la misma.

X	8 bits	8 bits	8 bits	8 bits	8 bits	8 bits	X
0	Pocicion / Velocidad		Senal de referencia		Accion de control		0

Tabla 3 Trama Micro controlador

3.3.3 Interfaz Gráfica

Labview es un entorno de programación gráfica en el cual se implemento la interacción del usuario con la plataforma.

Este instrumento cuenta con dos ventanas principales, el panel frontal y el diagrama de bloques. El panel frontal es la herramienta con la cual el usuario se va a comunicar, en esta están presentes los diferentes botones, indicadores y gráficas. En el diagrama de bloques es donde se realizo toda la programación y están los bloques que fueron implementados para desarrollar la aplicación.

3.3.3.1 Panel Frontal

En el panel frontal se encuentran los elementos que logran que el usuario interactué con la plataforma. En la figura se pueden encontrar los diferentes controles, indicadores y gráficas que se implementaron para el desarrollo de la misma.

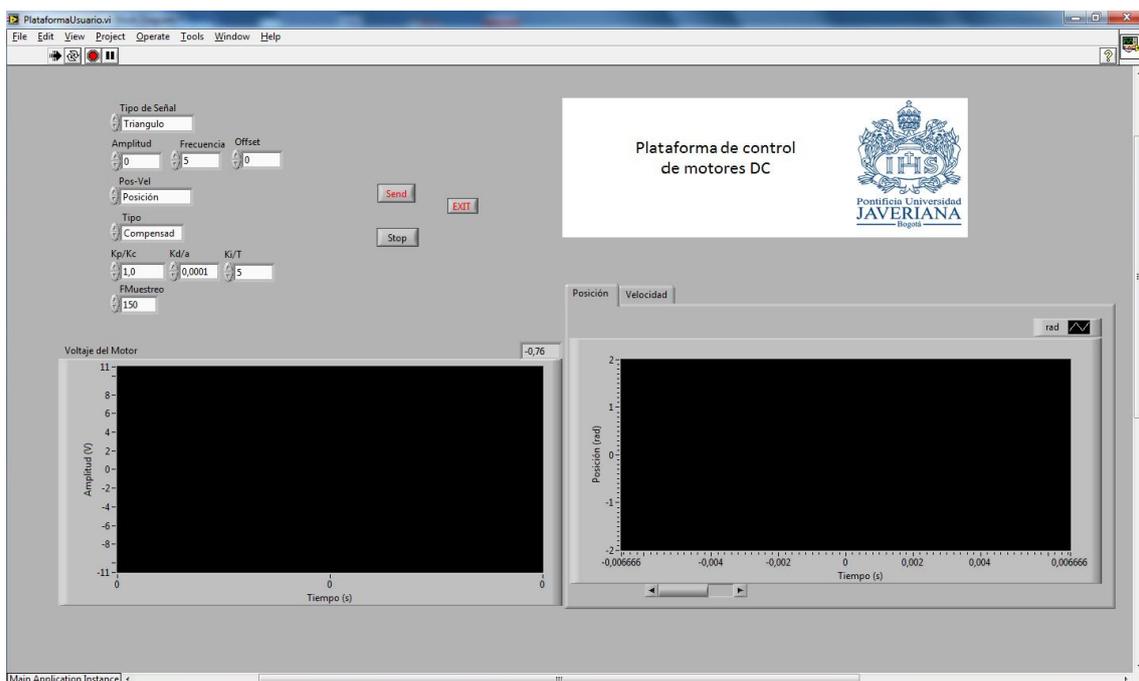


Figura 19 Panel Frontal

Los controles están destinados para ingresar los datos, todos estos permiten al usuario modificar los datos que posteriormente serán enviados y reconocidos. Estos controles son de dos tipos, Text ring y number control. Los controles de tipo text ring permiten asignar un valor numérico al texto que se encuentra en cada posición, de esta manera la señal cuadrada, será enviada como un '1', la triangulo como un '2' y el seno un '3', así mismo el proceso se repetirá para el tipo de control. Los controles numéricos registran el numero que se les digita, teniendo como limitante la representación asignada al control y los limites que se tienen para las variables según lo establecido en las especificaciones.

Aparte de los controles numéricos la interfaz gráfica cuenta con tres botones, SEND, STOP y EXIT. El botón STOP es el encargado de detener la marcha del motor, cada vez que se van a ingresar nuevos datos con el fin de realizar alguna modificación en los parámetros es necesario parar el motor. SEND se encarga de enviar los datos una vez han sido ingresados en su totalidad. El botón de EXIT termina toda la rutina y acaba el programa.

Se dispuso de tres gráficas para presentar los resultados de las experiencias, para el caso de los datos de posición o velocidad del motor se dispuso una tabla de donde es posible escoger que gráfica se desea visualizar dependiendo del caso, adicionalmente en el mismo gráfico se puede encontrar la señal de referencia que fue usada durante la acción de control. Otra gráfica de tipo arreglo fue usada para presentar el voltaje aplicado al motor. A las tres gráficas se les realizaron los ajustes necesarios, como se comentó en el diagrama de bloques, adicional a esto cada una tiene configurada una escala acorde para poder visualizar mejor los datos y el eje del tiempo fue ajustado en segundos.

3.3.3.2 Diagrama de Bloques

El diagrama de bloques se divide en dos partes principales, las cuales giran en torno a la tarea principal que se desempeña en la aplicación, la recepción y la transmisión de datos. Para esta tarea se implementó el bloque de control VISA (Virtual Instrument Software Architecture) con el cual es posible tener acceso al puerto serial.

Como primera medida se configura el tipo de comunicación con “VISA configure serial port” es con este bloque que se definen las características como, Baud rate, paridad, cantidad de bits y bits de parada. Estos fueron previamente definidos por el tipo de comunicación que se eligió y por las características de el microcontrolador.

Luego de definir los parámetros para la comunicación en el instrumento VISA, se utilizaron otras dos de las herramientas para el puerto serial, VISA Write y VISA Read. A través de estos dos bloques fluyen los datos dentro de la aplicación ya que constantemente se van a estar escribiendo y leyendo los datos en el puerto serial.

Para VISA Write es necesario conectar los parámetros previamente establecidos en la configuración de la sesión de visa, como el resource name, que hace referencia al nombre del puerto, en nuestro caso COM1. Luego los datos son enviados al puerto en un buffer de tipo string, razón por la cual fue necesario organizar los datos en un string, para ser reconocidos por la función. Para esto se formó un arreglo con todos los datos que serán ingresados por el usuario con la función Build Array y este arreglo se convierte en el string que posteriormente será entregado al puerto mediante la función Byte Array to String.

El control de la acción de escritura en el puerto serial se hace mediante una función condicional, que se representa como un case. Esta función ejecuta un subdiagrama dependiendo del valor ingresado en su entrada, cuando la etiqueta se encuentra en su estado verdadero los datos van a fluir hacia la sesión de visa. Al input del condicional se encuentran conectados dos botones y una función booleana, los cuales van a servir para enviar los datos una vez han sido ingresados y para enviar el dato de parada del motor al micro controlador, estos son los botones SEND y STOP, los cuales serán constantemente utilizados por el usuario, ya que cada vez que se requieran enviar datos nuevos es necesario detener la marcha del motor con el botón de STOP y luego con el botón SEND reiniciar la misma con el nuevo paquete de información. En la figura 20 se puede observar el diagrama de flujo para este proceso.



Figura 20 Diagrama de flujo Write

El siguiente bloque principal se desarrolla junto al VISA read, este va a ser el encargado de leer los datos que provienen del microcontrolador, con la información acerca de la acción de control. Este bloque también debe ser configurado con el resource name y los datos de la sesión de VISA previamente inicializada. El diagrama de flujo para este proceso, se puede observar en la figura 21.

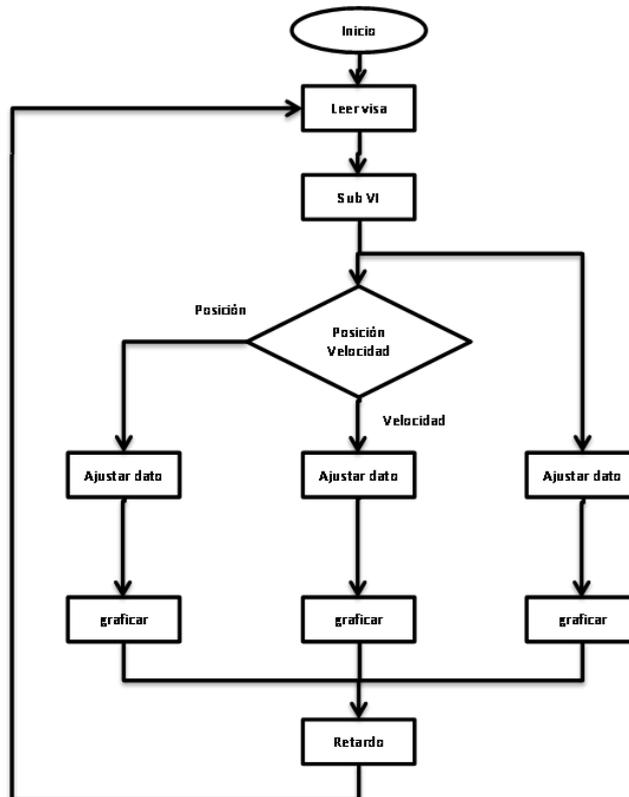


Figura 21 Diagrama de Flujo Read

Como se observa en la Figura 21, la sesión de VISA almacena los datos en el buffer de lectura, es a partir de este que se toma la palabra enviada desde el micro controlador. Como se describió anteriormente los datos son enviados en una trama de 64bits en donde se encuentran el encabezado, la cola y los datos correspondientes a velocidad o posición del motor, voltaje y la señal de referencia que fue utilizada la acción de control.

Así pues los datos una vez en el buffer deben ser tratados, para identificar el encabezado y cola y garantizar que la palabra llegó en el orden deseado y visualizar las variables esperadas. Para realizar esta tarea se configuro un instrumento virtual que contiene la rutina que recibe el string y que retorna cada variable lista para ser graficada.

Como se observa en la figura 22, este sub instrumento virtual toma el string y lo convierte en un arreglo, en donde cada posición corresponde a un dato de la palabra. Para esto emplea la función string to byte array

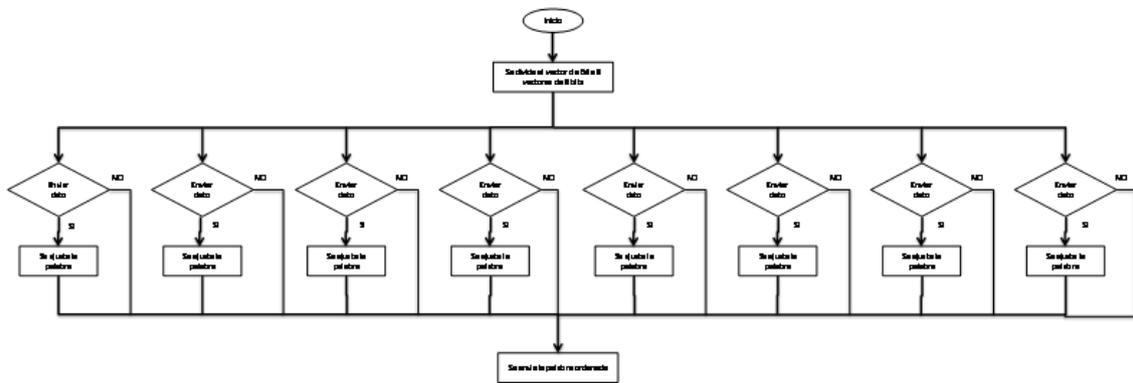


Figura 22 Diagrama de Flujo SubVI

Luego de este procedimiento es necesario separar la información del arreglo y reconocer como se recibió el paquete. La función array subset es la encargada, esta función recorre todo el arreglo hasta la posición indicada en el index y extrae la cantidad de bits indicados en el length, este proceso divide la palabra en 8 paquetes de 8 bits. Conforme se reciben la cabeza y la cola que se representan por el carácter '0' se organizan los datos según se muestra en la tabla 5.

Posición	1	2	3	4	5	6	7	8
Opción 1	0	A1	A2	B1	B2	C1	C2	0
Opción 2	A1	A2	B1	B2	C1	C2	0	0
Opción 3	A2	B1	B2	C1	C2	0	0	A1
Opción 4	B1	B2	C1	C2	0	0	A1	A2
Opción 5	B2	C1	C2	0	0	A1	A2	B1
Opción 6	C1	C2	0	0	A1	A2	B1	B2
Opción 7	C2	0	0	A1	A2	B1	B2	C1
Opción 8	0	0	A1	A2	B1	B2	C1	C2

Tabla 4 Diagrama Datos

Como se puede observar en la Tabla 5, los datos pueden llegar ubicados en las diferentes 8 opciones, la rutina reconoce cual de las opciones es la que fue leída en el puerto serial, comparando y encontrando la posición de los ceros en el arreglo, con esto los datos en las otras posiciones pasan a un condicional que en su estado verdadero forman los datos que serán gráficos posteriormente.

Para las gráficas se emplearon cuatro indicadores gráficos, antes de ser conectados, los datos fueron ajustados para que estos coincidieran con las unidades correspondientes a cada variable.

4. Resultados

A continuación se presentarán, los resultados que muestran el correcto funcionamiento de la plataforma y los resultados de algunos de los diferentes experimentos que podrán ser realizados con la misma.

4.1 Frecuencia de Muestreo

Como se mencionó anteriormente la plataforma cuenta con tres diferentes frecuencias de muestreo, que se generan desde el micro controlador a partir de la interrupción del timer 3. Desde la interfaz se seleccionaron cada una de las frecuencias posibles y se visualizaron, para comprobar que el sistema se encuentra trabajando a la frecuencia deseada.

En la gráfica se puede observar la primer frecuencia que corresponde a 150Hz.

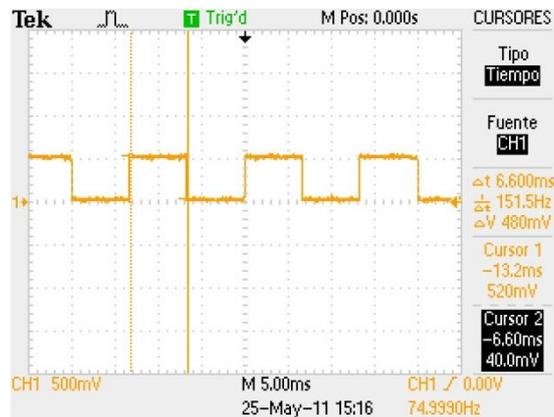


Figura 23 Frecuencia de muestreo 150Hz

La siguiente frecuencia es múltiplo entero de la primera y es de 75.76Hz.



Figura 24 Frecuencia de muestreo 75Hz

Por último la frecuencia de 37Hz



Figura 25 Frecuencia de muestro 37Hz

Con esto se garantiza que las frecuencias de muestreo corresponden a las escogidas por el usuario. En cada uno de los tres casos.

4.2 Tiempo de Control

Este es el tiempo que le toma al micro controlador, para calcular la acción de control, como se puede observar es de 168µs con lo que se garantiza que este es menor a la frecuencia de muestreo.

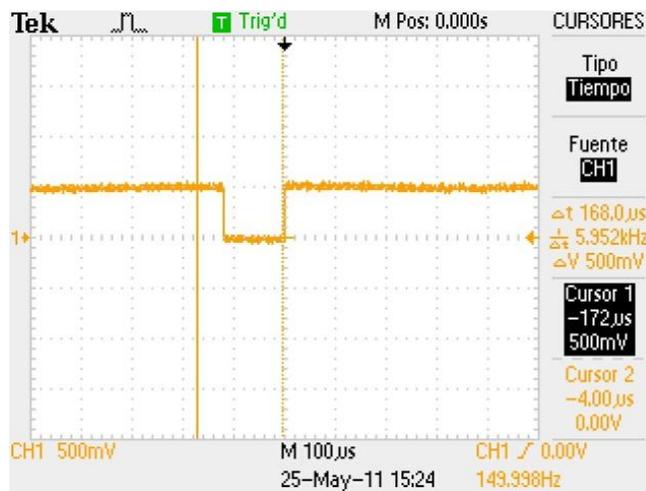


Figura 26 Tiempo accion de control

4.3 Modelado

Para diseñar el control de un motor DC es necesario realizar un modelado de este componente y encontrar su función de transferencia, para lograr este objetivo se debe colocar el control de la interfaz en malla abierta, seleccionar una señal cuadrada de amplitud 0 y variar el offset de 1 a 10 con intervalos de un voltio. En cada medida se deben tomar los siguientes datos, corrientes del motor y velocidad. La corriente debe ser medida con un amperímetro en serie con la fuente que alimenta el micro controlador tal como se muestra en la figura 27, es necesario medir también la corriente que se obtiene cuando el motor no está en movimiento y restar la misma a los valores obtenidos posteriormente. El dato de la velocidad se puede visualizar en la interface.

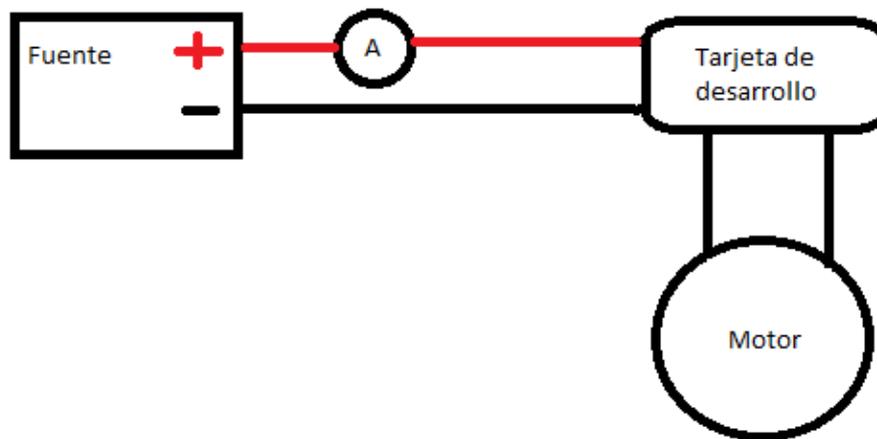


Figura 27 Medición Corriente

Los datos que se obtuvieron en las medidas se pueden observar en la siguiente tabla, en base a la tabla se obtuvo un promedio del valor de la resistencia R_m y de la constante K_m :

Voltaje (V)	Encoder (Δ pulsos)	Angulo (grados)	Angulo (rad)	Velocidad (rad/s)	Corriente (mA)	CorrienteMax (mA)	R_m (Ω)	K_m (Nm/A)
10	366	128,671875	2,245747873	336,8622147	46	970	10,3092784	0,02968573
9	330	116,015625	2,02485464	303,7282264	45	900	10	0,02963175
8	290	101,953125	1,779417714	266,9126838	43	820	9,75609756	0,02997235
7	254	89,296875	1,55852448	233,7786955	42	730	9,5890411	0,02994285
6	215	75,5859375	1,319223478	197,8835414	41	625	9,6	0,03032086
5	179	62,9296875	1,098330244	164,7495531	40	522	9,57854406	0,0303491
4	137	48,1640625	0,840621472	126,0932334	39	418	9,56937799	0,03172256
3	103	36,2109375	0,632000085	94,80002217	38	318	9,43396226	0,03164556
2	62	21,796875	0,380427235	57,06409102	37	209	9,56937799	0,03504831
1	17	5,9765625	0,104310694	15,6466056	35	105	9,52380952	0,06391163
							9,69294888	0,03422307

Tabla 5 Modelado Motor

La función de transferencia del motor en el dominio de Laplace está dada de la siguiente forma:

$$G_{\theta}(s) = \frac{K}{s(\tau s + 1)}$$

Dónde:

$$K = \frac{1}{0.034}$$

Remplazando los valores previamente mencionados obtenemos:

$$K = \frac{1}{0.034}$$

$$K = 29.4 \frac{Nm}{A}$$

El τ del motor se obtuvo mediante la respuesta de la planta ante una entrada paso tal como se puede observar en la Figura 28.

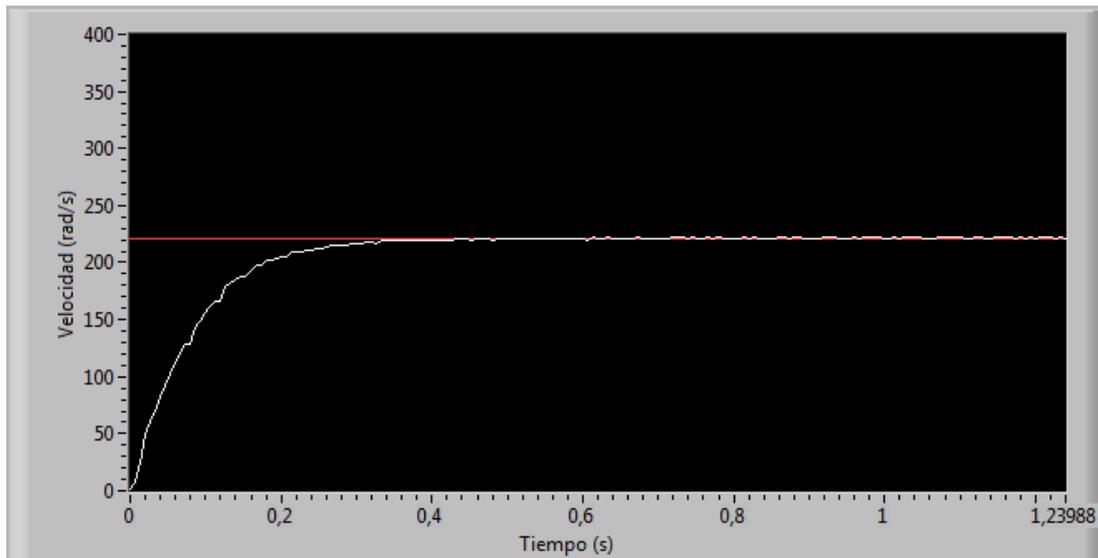


Figura 28 Respuesta Paso del Motor

$$\tau = 160 \text{ ms}$$

Como resultado la función de transferencia de la planta es:

$$G_{\theta}(s) = \frac{29.4}{s(0.16s + 1)}$$

4.4 Control PID de posición

Mediante la herramienta sisotool de matlab se sintonizó el controlador y se obtuvieron las siguientes ganancias:

$$K_p = 1$$

$$K_d = 0.01$$

$$K_i = 5$$

Con la ayuda de simulink se simuló la respuesta del sistema a estas ganancias, como se puede apreciar en la figura 28:

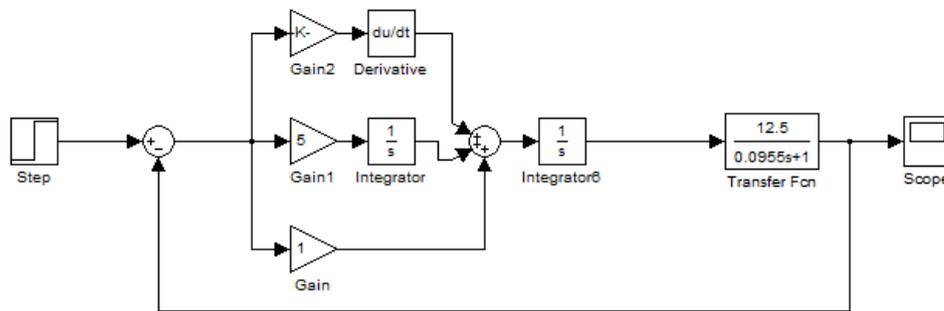


Figura 29 Diagrama de Bloques Control de Posición

El resultado obtenido para esta simulación se pueden observar en la figura 29, esta muestra un sobre piko del 50%.

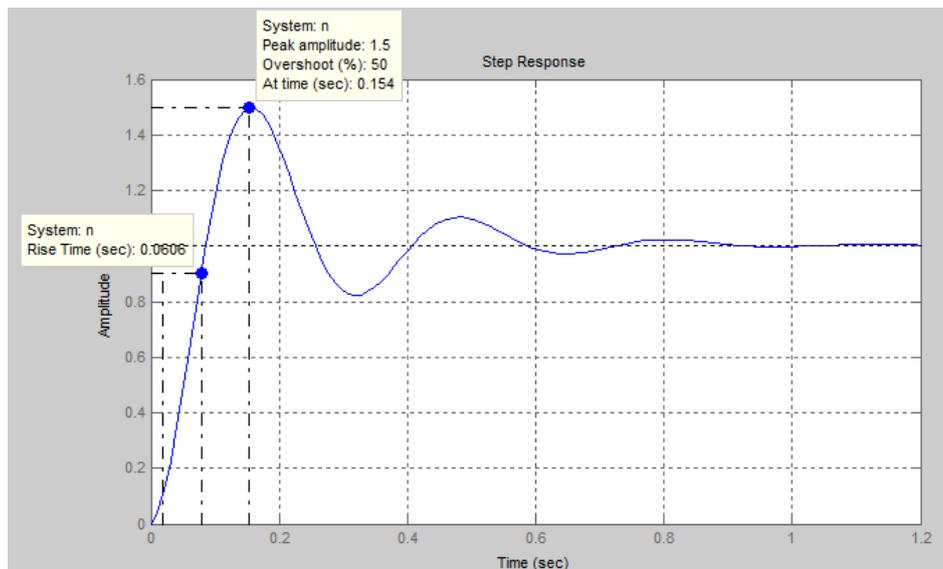


Figura 30 Simulación Control PID posición

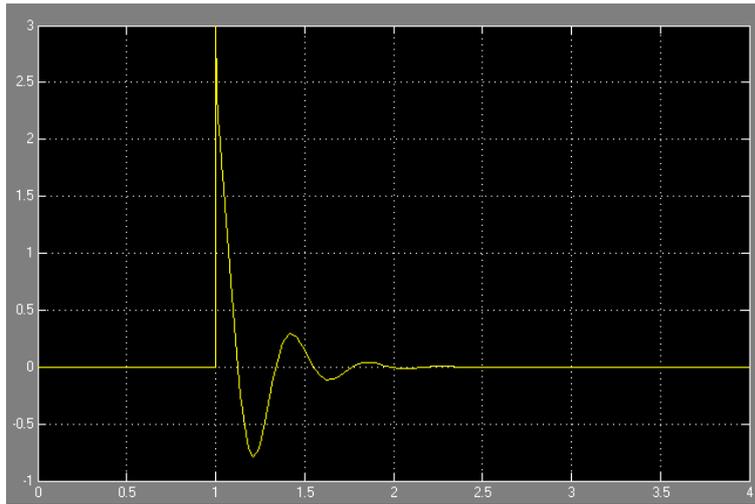


Figura 31 Simulación Voltaje aplicado a la planta

En la figura 31 se puede observar el resultado entregado por la interfaz, donde se aprecia un sobre pico del 70% y una forma de onda similar a la obtenida en la simulación, en la figura 31 se encuentra el voltaje del motor, este se encuentra de los límites de la fuente de alimentación por lo tanto el control no está saturado.

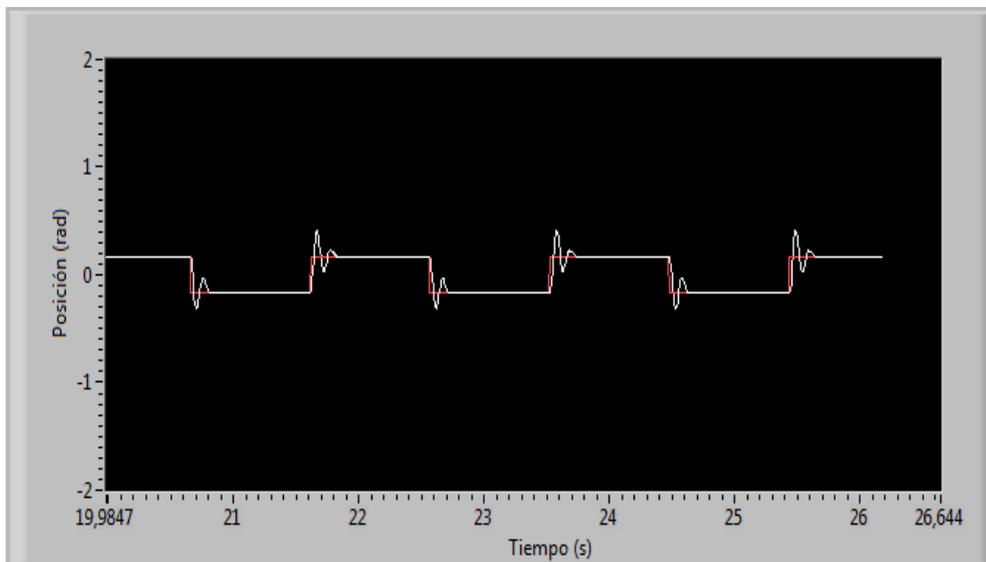


Figura 32 Control Posición Interfaz

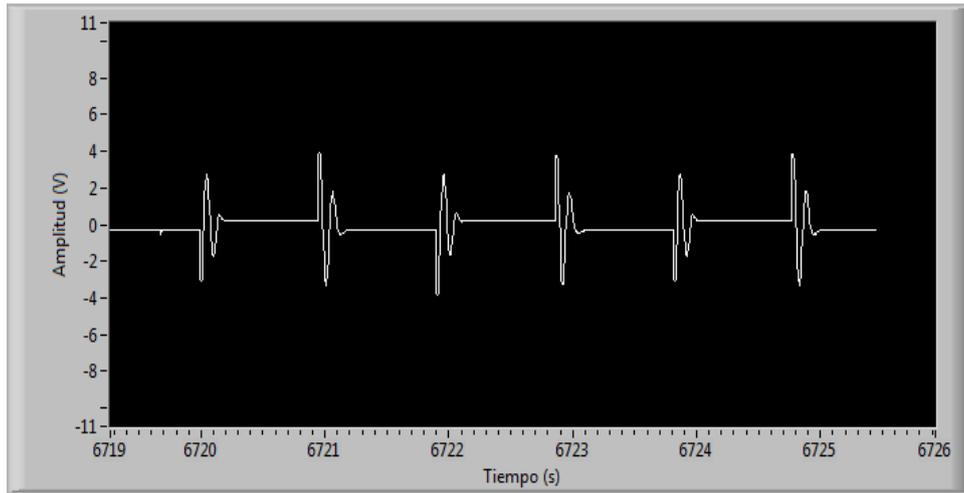


Figura 33 Voltaje del Motor

Al utilizar los mismos parámetros en la interfaz para una forma de onda triangular tal como se puede apreciar en la figura 34, se aprecia una respuesta por parte del sistema. En la figura 35 podemos observar el voltaje de alimentación del motor.

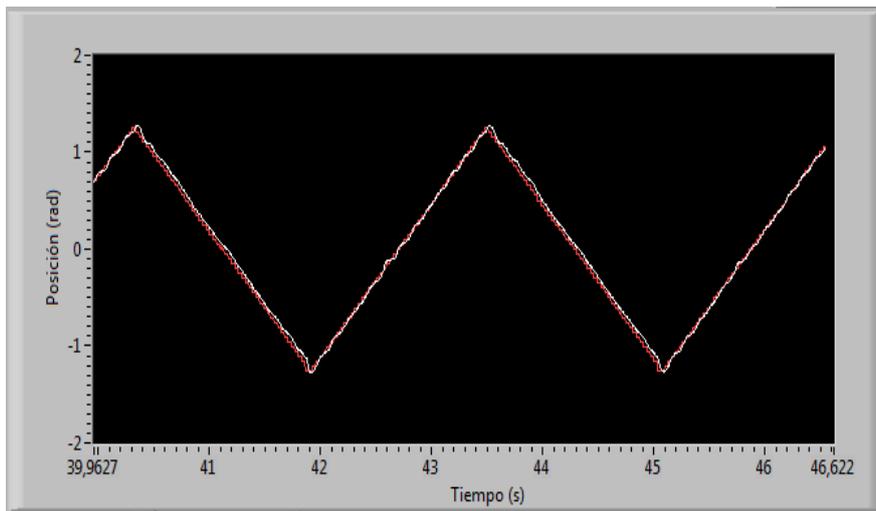


Figura 34 Control de Posición Referencia Triangulo

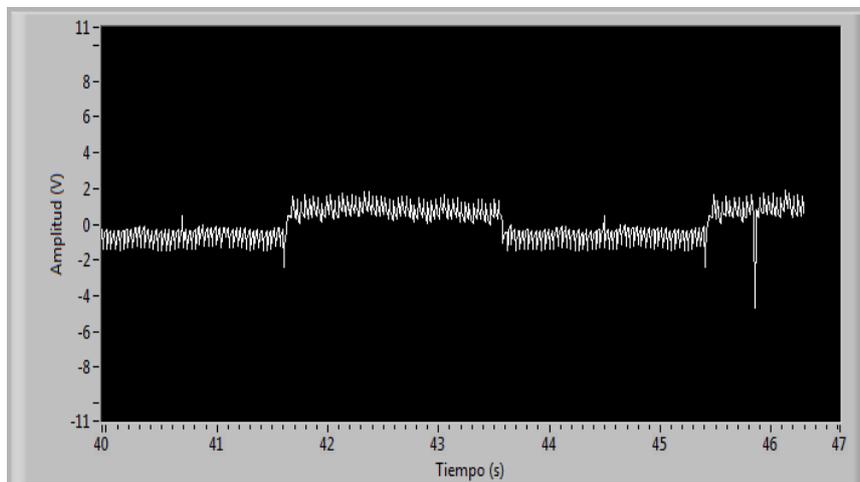


Figura 35 Voltaje del Motor Referencia Triangulo

4.5 Control Compensador Posición

Estos son los resultados para un compensador en atraso, con los siguientes parámetros.

$$\alpha=0.0001$$

$$\tau=5$$

$$K= 1$$

En la Figura 36 se puede observar como el control responde ante una señal de referencia cuadrada.

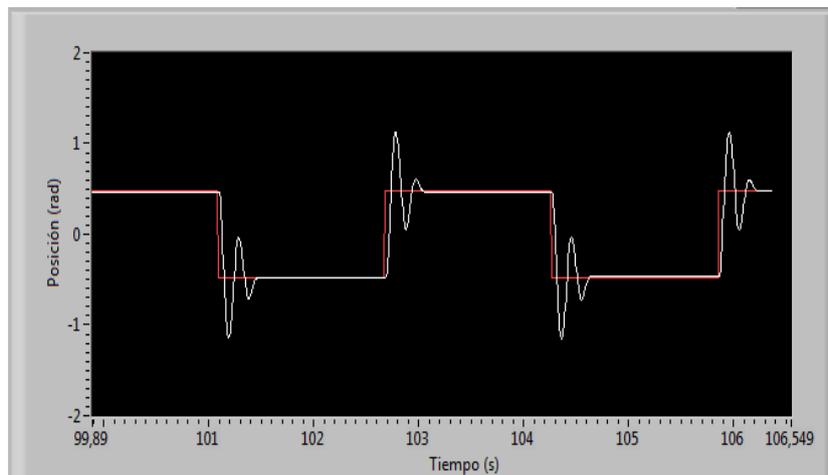


Figura 36 Compensador Posición

En la figura 37 se encuentra el voltaje aplicado al motor durante la acción de control.

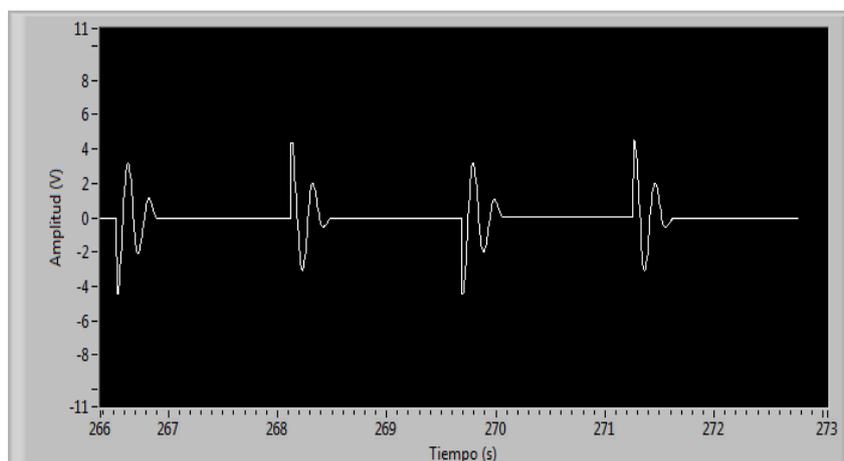


Figura 37 Voltaje Motor

En la figura 38 se puede observar como el compensador reacciona ante una señal de referencia triangulo.

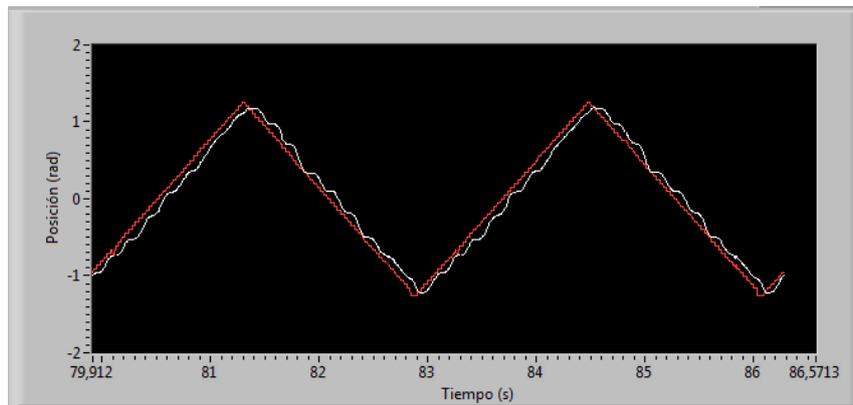


Figura 38 Compensador Posición Triangulo

En la figura 39 se puede observar el voltaje aplicado al motor, durante la acción de control.

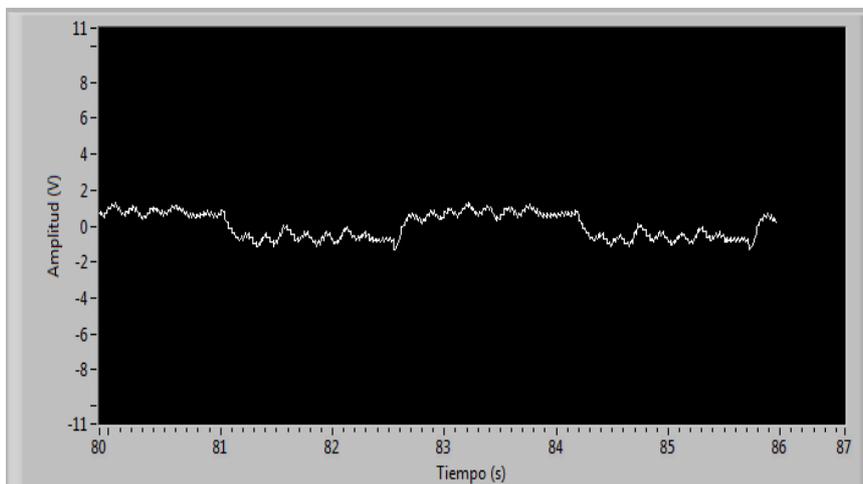


Figura 39 Voltaje Motor

4.6 Control PID de Velocidad

Conociendo la función de transferencia del motor y la del controlador se puede obtener el diagrama de bloques con la realimentación del sistema, tal como se puede observar en la figura 40.

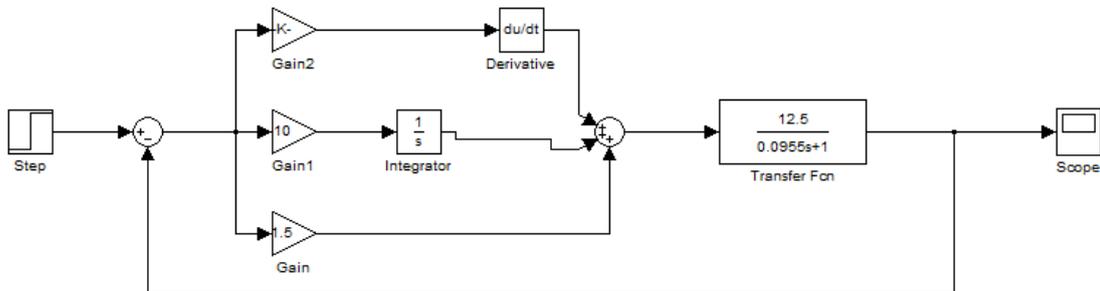


Figura 40 Diagrama de bloques Control de Velocidad

Utilizando algebra de bloques se obtiene la función de transferencia del sistema y está dado por la siguiente expresión:

$$G_S(s) = \frac{G_\theta(s)G_c(s)}{G_\theta(s)G_c(s) + 1}$$

De donde:

$$G_c(s) = \frac{K_p * s + K_i + K_d * s^2}{s}$$

$$G_\theta(s) = \frac{K}{s(\tau s + 1)}$$

Luego al remplazar las ecuaciones:

$$G_S(s) = \frac{s^2 K_d K + s K K_p + K_i K}{s^2(\tau + K_d K) + s(K K_p + 1) + K_i K}$$

$$G_S(s) = \frac{1}{\tau + K_d K} * \frac{s^2 K_d K + s K K_p + K_i K}{s^2 + s \frac{K K_p + 1}{\tau + K_d K} + \frac{K_i K}{\tau + K_d K}}$$

Mediante la ecuación de un sistema de segundo orden se obtiene:

$$2\xi\omega_0 = \frac{K K_p + 1}{\tau + K_d K}$$

$$\omega_0^2 = \frac{K_i K}{\tau + K_d K}$$

Se diseñó un control con un sobre pico (PO) del 20% y un tiempo de pico (t_p) de 0.2 s

$$PO = 100e^{\frac{-\pi\xi}{\sqrt{1-\xi^2}}}$$

$$\xi = 0.456$$

Luego de remplazar el valor de ξ y t_p en la siguiente ecuación obtenemos ω_0

$$\omega_0 = \frac{\pi}{t_p\sqrt{1-\xi^2}}$$

$$\omega_0 = 17.64$$

Con estos valores y una ganancia proporcional $K_p = 1$ de 1 se obtienen K_i y K_d :

$$K_i = 20.7$$

$$K_d = 0.06$$

El resultado obtenido en la interfaz se puede observar en la Figura 41. Donde el sobre pico es del 30% y el tiempo pico es de aproximadamente 0.2 segundos. La figura 42 muestra el voltaje del motor.

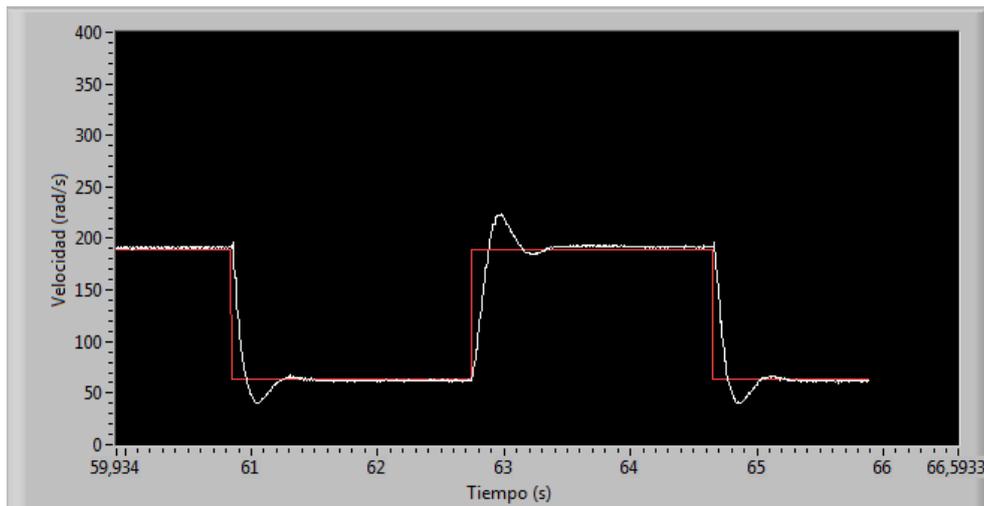


Figura 41 Control Posición Interfaz

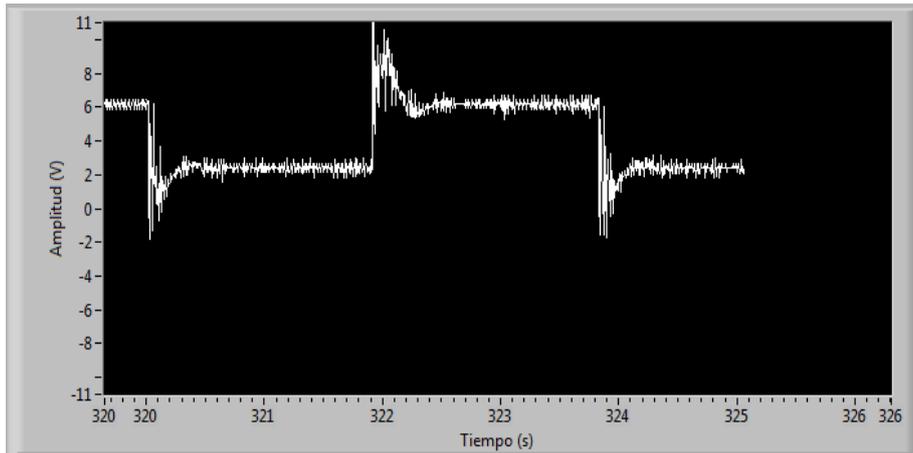


Figura 42 Voltaje Motor

Al cambiar la señal de referencia por una seno podemos apreciar que la velocidad del motor cambia y sigue la señal de referencia con precisión y a su vez en la figura 43 podemos ver como varía el voltaje del motor.

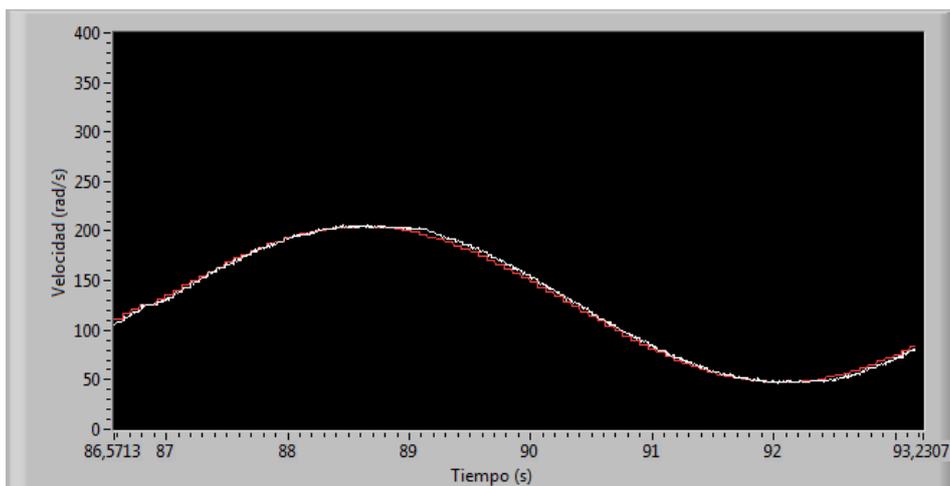


Figura 43 Control de velocidad referencia Seno

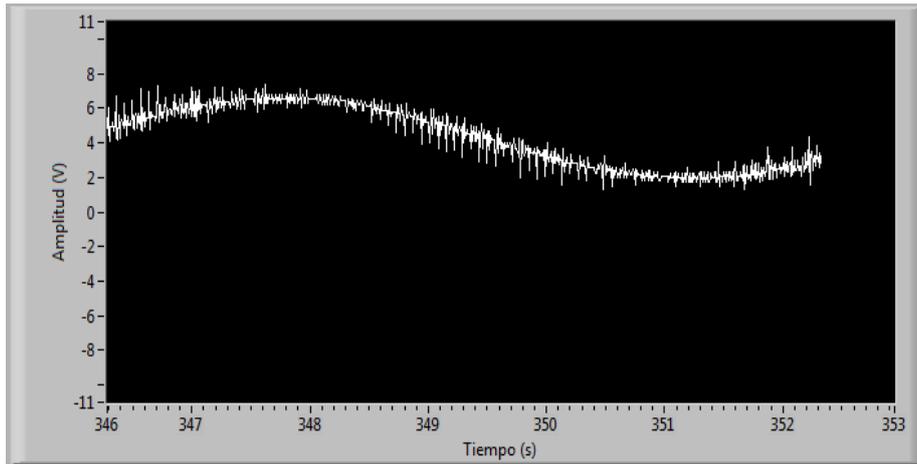


Figura 44 Voltaje del motor referencia Seno

5. Conclusiones

La plataforma de control es un sistema que permite a su usuario controlar un motor DC de forma interactiva, a través de una interfaz gráfica amable con el usuario. Esta plataforma puede ser incluida como parte de un componente experimental, de los cursos de controles y sistemas dinámicos, para estimular y fomentar el interés en estas asignaturas, debido a que les brinda la posibilidad de poner en práctica los temas vistos en clase. Por medio de la interfaz gráfica es posible validar el diseño de la estrategia de control elegida por el usuario. Ofreciendo la posibilidad de modificar los parámetros calculados con el fin de mejorar la respuesta del sistema. Haciendo de este proyecto una herramienta que aporta al aprendizaje de los estudiantes.

Esta plataforma responde satisfactoriamente a los controles PID de posición y velocidad diseñados y puestos a prueba por el usuario, igualmente a los experimentos que se realizan en malla abierta. Permitiendo observar las variables implicadas en este proceso, tales como la señal de referencia, el voltaje del motor y la acción de control realizada, en tiempo real.

El protocolo de comunicación seleccionado no fue el más óptimo para todas las estrategias de control, ya que debido a la limitación que se impone por el número de bits permitidos, el rango de valores posibles para la sintonización de los controladores no es muy amplio. De la misma forma durante la transmisión de datos se presenta pérdida de la información, alterando levemente la observabilidad de los mismos. Con la mejora de este protocolo se podrá optimizar la respuesta del compensador en atraso que hace parte de la plataforma, esto también abrirá la puerta para más implementaciones que permitirán ampliar la funcionalidad de esta plataforma.

6. Bibliografía

- [1] Karl Johan Astrom And Jacob Apkarian, Haerve Lancheray. USB QICii Laboratory Workbook. Document Number – revision 01
- [2] Villato F., Rodrigez A. *“Diseño e implementación de un laboratorio para la enseñanza de sistemas de control”*. Pontificia Universidad Javeriana. 2003.
- [3] DORF Richard and BISHOP Robert. *MODERN CONTROL SYSTEMS - 10th EDITION* Upper Saddle River NJ: Pearson Prentice Hall, 2005.
- [4] OGATA Katsuhiko. *INGENIERÍA DE CONTROL MODERNA*. 4ta. Edición. Madrid: Pearson Prentice Hall. 2003.
- [5] Sección de Control. *“Programa de la asignatura Sistemas Dinamicos”*. 2010
- [6] Sección de Control. *“Programa de la asignatura Controles”*. 2010
- [7] Microchip. *“Family Reference Manual”*. 2006

- [8] OGATA Katsuhiko. *SISTEMAS DE CONTROL EN TIEMPO DISCRETO*: Pearson Prentice Hall- 1996

ANEXOS

A. Código Microcontrolador

Se encuentra adjunto en el CD

B. Script Matlab función Seno

Se encuentra adjunto en el CD

C. Plataforma del Usuario

Se encuentra adjunto en el CD

D. Manual Del Usuario

Se encuentra adjunto en el CD

Índice de Figuras

Figura 1 Diagrama de Flujo Plataforma.....	7
Figura 2 Diagrama de bloques Plataforma.....	9
Figura 3 Sistema de control PID.....	10
Figura 4 dsPIC30f2010.....	14
Figura 5 Motor Pittman.....	14
Figura 6 Funcionamiento Puente H.....	15
Figura 7 Conteos PWM.....	15
Figura 8 Diagrama de bloques Encoder.....	16
Figura 9 Rango Dinámico Posición.....	16
Figura 10 Rango Dinámico Velocidad.....	17
Figura 11 Diagrama de flujo Main.....	20
Figura 12 Diagrama de Flujo U1RX1Interrupt.....	21
Figura 13 Diagrama de flujo Cuadrada.....	22
Figura 14 Diagrama de flujo T3Interrupt.....	23
Figura 15 Diagrama de flujo Generadorfunciones.....	24
Figura 16 Diagrama de flujo EnviarDato.....	25
Figura 17 Diagrama de flujo Seno.....	26
Figura 18 Diagrama de flujo Triangulo.....	27
Figura 19 Panel Frontal.....	30
Figura 20 Diagrama de flujo Write.....	32
Figura 21 Diagrama de Flujo Read.....	32
Figura 22 Diagrama de Flujo SubVI.....	33
Figura 23 Frecuencia de muestreo 150Hz.....	35
Figura 24 Frecuencia de muestreo 75Hz.....	35
Figura 25 Frecuencia de muestro 37Hz.....	36
Figura 26 Tiempo accion de control.....	36
Figura 27 Medición Corriente.....	37
Figura 28 Respuesta Paso del Motor.....	38
Figura 29 Diagrama de Bloques Control de Posición.....	39
Figura 30 Simulación Control PID posición.....	39
Figura 31 Simulación Voltaje aplicado a la planta.....	40
Figura 32 Control Posición Interfaz.....	40
Figura 33 Voltaje del Motor.....	41
Figura 34 Control de Posición Referencia Triangulo.....	41
Figura 35 Voltaje del Motor Referencia Triangulo.....	41
Figura 36 Compensador Posición.....	42
Figura 37 Voltaje Motor.....	42
Figura 38 Compensador Posición Triangulo.....	43
Figura 39 Voltaje Motor.....	43
Figura 40 Diagrama de bloques Control de Velocidad.....	44
Figura 41 Control Posición Interfaz.....	45
Figura 42 Voltaje Motor.....	46
Figura 43 Control de velocidad referencia Seno.....	46
Figura 44 Voltaje del motor referencia Seno.....	47

