

CONTROL DE LA VELOCIDAD EN TIEMPO REAL DE UN MOTOR DC CONTROLADO POR LÓGICA DIFUSA TIPO PD+I USANDO LABVIEW

REAL –TIME DC MOTOR VELOCITY CONTROL BY FUZZY LOGIC TYPE PD+I CONTROLLERS USING LABVIEW

Ricardo Rodríguez Bustinza¹, Ewar Mamani Churayra²

RESUMEN

En este artículo, se presenta un método basado en inteligencia artificial para controlar una planta motor DC por un microordenador personal (PC), el que interactuando hardware y software logra el control de la velocidad del motor DC en tiempo real usando el algoritmo de control Difuso-PD+I. La adquisición de datos e identificación de los parámetros del motor DC han sido necesarias para el control de la velocidad del motor DC, por medio de la tarjeta de adquisición de datos PCI NIDAQ 6024E cuya interface corre en tiempo real que usa el Workshop Real-Time (RTW), el archivo de datos es procesado con la herramienta de identificación del programa Matlab llamada IDENT. El prototipo del sistema computadora-controlador se diseña empleando la programación grafica de LabVIEW, en este caso se hace uso de las herramientas Fuzzy Logic Control y Simulation Module. El control en tiempo real del sistema se lleva a cabo en el laboratorio usando el convertidor digital-a-analógico (DAC) y encoder formado por dos sensores de efecto hall de tipo incremental que por medio de un convertidor frecuencia voltaje se logra procesar las señales desde las entradas analógicas de la NIDAQ. Se verifican los resultados de simulación de computadora experimentalmente, los que demuestran que la señal de control diseñada puede hacer que la salida del sistema prototipo siga eficientemente las referencias impuestas con mínimo sobrepaso y error en estado estacionario nulo.

Palabras clave.- Motor DC, Adquisición de datos, Identificación de parámetros, Diseño del controlador e implementación.

ABSTRACT

In this article, a method is presented based on artificial intelligence to control a plant DC motor for a personal microcomputer (PC), that interacted hardware and software achieves the control of the speed of the DC motor in real time using the control algorithm Fuzzy-PD+I. The acquisition of data and identification of the parameters of the DC motor have been necessary for the control of the speed of the motor DC, by means of the card of acquisition of data PCI NIDAQ 6024E whose interface runs in the real time that the Workshop Real-Time uses (RTW), the file of data is processed with the tool of identification of the program called IDENT of Matlab. The prototype of the system computer-controller is designed using the graphic programming of LabVIEW, in this case use of the tools Fuzzy Logic Control and Simulation Module. The control in real time of the system is carried out in the laboratory using the digital-to-analogical converter (DAC) and incremental encoder formed by two sensors of effect hall that is possible to process the signs from the analogical input of the NIDAQ by means of a convertor frequency voltage. The results of computer simulation are verified experimentally, those that demonstrate that the designed control sign can make that the exit of the system prototype follows the references imposed with minimum overshoot and null steady-state error.

¹M.Sc, Ing. Profesor Investigador de la Facultad de Ingeniería Mecánica de la Universidad Nacional de Ingeniería, ²Bachiller Investigador de la Escuela Profesional de Ingeniería Mecatrónica, Universidad nacional de Ingeniería.

Keywords.- DC Motor, Data acquisition, Parameters identification, Control design and implementation.

INTRODUCCION

El “hacer girar” a los motores, es una tarea complicada, más aun cuando estos están siendo controlados por una computadora. El propósito de este artículo es presentar un método práctico y sistemático para el diseño, simulación, y control basado en lógica difusa en tiempo real de una planta de motor DC, usando una computadora personal como el organizador y desarrollador de objetivos mediante el programa gráfico de LabVIEW.

Hay muchos tipos de motores DC en la industria que usan una inercia del rotor que pueden ser muy pequeñas y como resultado comercialmente existen, motores con muy alto torque-inercia. Los sistemas de motores DC son generalmente controlados por la técnica convencional de controladores Proporcional-Integral-Derivativo (PID), debido a que ellos son diseñados fácilmente, tienen un bajo costo, el mantenimiento es barato y con alta efectividad. A menudo es necesario saber el modelo matemático para constituir algunos experimentos para sintonizar los parámetros del controlador PID. Sin embargo, se conoce que los controladores PID convencionales generalmente no trabajan bien para los sistemas no-lineales, particularmente los sistemas complejos con incertidumbres y que no tienen ningún modelo matemático preciso. Para superar estas dificultades, hay varios tipos de modificaciones de controladores PID convencionales como el desarrollo de controladores PID adaptivos, [1].

También el controlador basado en lógica difusa (FLC) puede usarse para este tipo de problemas. Cuando se compara al controlador convencional, la ventaja principal de la lógica difusa es que no requiere del modelo matemático.

Desde que el controlador presenta base de reglas difusas basadas en el conocimiento del sistema y la experiencia del ingeniero de control, el FLC requiere un modelo matemático menos complejo que los controladores convencionales. Así mismo, para lograr una alta performance los FLC necesitan un esquema alternativo para el control de un sistema, en este caso se trata de un controlador Difuso-PD+I cuyo diagrama de bloques se muestra

en la Fig. 1, donde observamos que la acción integral es externa al sistema, siendo beneficioso cuando se reduce el offset.

El diagrama también presenta los bloques que interactúan con las salidas y entradas analógicas de la NIDAQ PCI 6024E.

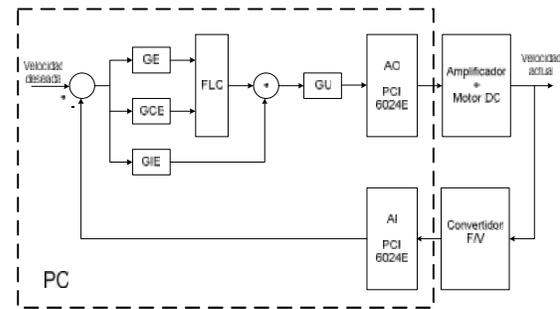


Fig. 1 Sistema de adquisición de datos.

ADQUISICION DE DATOS

La adquisición de datos es el proceso que involucra la recopilación de información de una forma automatizada a partir de fuentes de medición análogas y digitales como sensores y dispositivos bajo prueba. La adquisición de datos utiliza una combinación de medición de hardware y software basado en PC para proporcionar un sistema de medición flexible y definido por el usuario.

El proceso a controlar es un Motor DC EMG30 que interconectada con el puerto PCI de la computadora donde se encuentra insertada la tarjeta de adquisición de datos PCI 6024E que corre en tiempo real, si combinamos la habilidad de la computadora digital de llevar a cabo cálculos complicados y realizar diversas decisiones de velocidad y exactitud, resultará un sistema muy flexible que puede manejar una gran variedad de problemas con el esfuerzo de una programación mínima.

La adquisición de los datos y diseño del controlador se llevará a cabo mediante las herramientas del lenguaje gráfico de LabVIEW que nos permite implementar el sistema de control, manejando las entradas/salidas análogas en forma conveniente como se muestra en la Fig. 2.



Fig. 2 Sistema de adquisición de datos.

El procedimiento experimental de la Fig. 2, nos conduce a obtener la DATA del sistema, la misma que será analizada para los fines de la identificación del proceso. Por identificación se entiende la formulación de un modelo que permita representar al sistema con todas sus propiedades y características relevantes. En general, si realizamos un experimento en que aplicamos ciertos valores a las entradas del sistema, obtendremos ciertas respuestas en sus salidas; de ese modo, dando los mismos valores que en el experimento a las variables independientes de nuestro modelo, podemos esperar que después de resolver el conjunto de ecuaciones, la respuesta del modelo debe tener respuestas similares a aquellas salidas del sistema físico. En nuestro caso usaremos el Toolbox de Identificación de parámetros de Matlab, que es una interface gráfica mostrada en la Fig. 3.

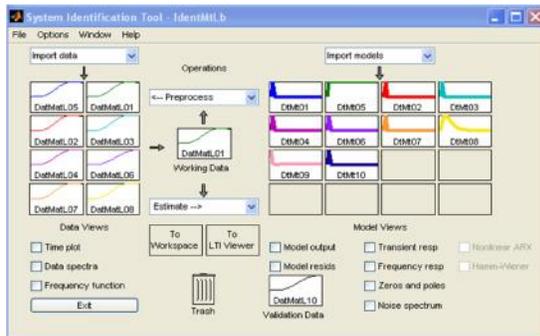


Fig. 3 Interfaz gráfica del toolbox de identificación (IDENT) de matlab.

IDENTIFICACIÓN DE PARÁMETROS

El procedimiento para obtener los parámetros del motor DC EMG-30, y en general de cualquier otro motor DC, presenta el siguiente procedimiento:

Primero.- Usaremos dos motores DC iguales (o con características similares) y que estén acoplados mecánicamente en los ejes de forma que un motor funcione como generador y el otro como

motor como se muestra en la Fig. 4. En el motor que funciona como generador las conexiones que debieran ser la alimentación del motor DC, funcionan como salida, siendo la respuesta del sistema a una entrada arbitraria, lo que quiere decir que la entrada en el motor que funciona como motor DC debe ser variable para apreciar la salida y/o respuesta en el motor que funciona como generador.



Fig. 4 Acople mecánico de los motores EMG30.

Segundo.- Las entradas análogas en una tarjeta de adquisición de datos son usadas para señales provenientes de cada motor, dichas señales corresponden a la alimentación de cada motor, la primera señal es la alimentación variable al motor que funciona como tal, y que en este caso es la variable de entrada, la segunda señal adquirida proviene de la alimentación del motor que funciona como generador, en este caso es la variable de salida, se debe tener en cuenta que debido a la simetría de los motores, la conexión de polaridad de la segunda señal hacia la tarjeta de adquisición de datos debe ser inversa a la que se realizó con la primera señal. Se miden directamente los voltajes de alimentación de los motores debido a la relación lineal existente entre la alimentación de un motor DC y la velocidad que estos desarrollan. Las señales adquiridas mediante Matlab (ver Fig. 5).

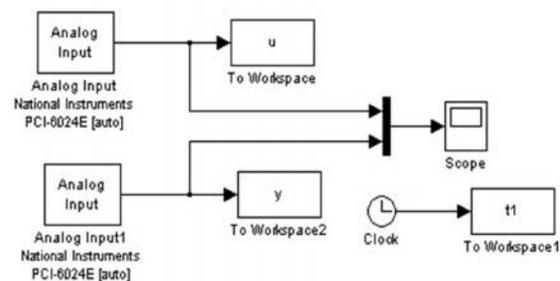


Fig. 5 Diagrama de bloques en Simulink para la adquisición de datos.

Tercero.- Se procede a adquirir la DATA para el pre procesamiento de los datos, para ello se cuenta con diferentes medidas que serán procesadas en diferentes archivos. Se hace necesario filtrar estas señales adquiridas, para esto se implemento un

filtro digital pasa bajo que anule la señal de ruido acoplada a estas señales. En la Fig. 6 se muestra un ejemplo de las señales originalmente adquiridas por la tarjeta PCI6024E antes y después del filtrado respectivo.

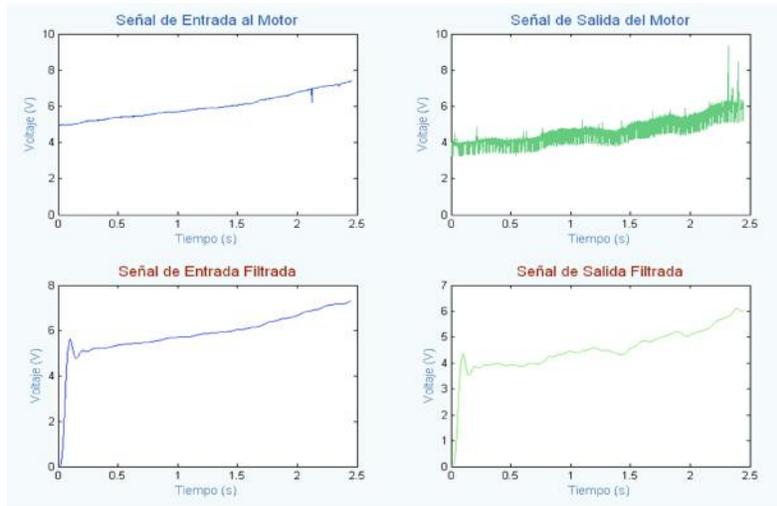


Fig. 6 Señales adquiridas desde la PCI 6024E.

Cuarto.- Generar la función de transferencia del Motor DC, usando los datos filtrados, mediante la herramienta IDENT de Matlab aproximamos una función de transferencia que represente el motor, esto requiere configurar adecuadamente la función de transferencia a la que se desea aproximar,

además IDENT permite comparar las respuestas de todas las funciones de transferencia aproximadas con cada entrada, es decir muestra y compara las respuestas de los diferentes sistemas aproximados para una misma entrada con la respuesta real tal como se muestra en la Fig. 7.

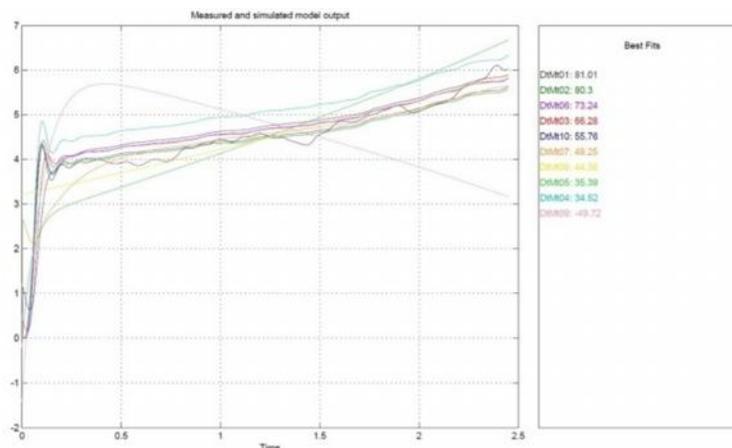


Fig. 7 Comparando las señales para procesamiento con el IDENT de matlab.

Luego de comparar todas las funciones de transferencia con cada uno de los modelos se obtiene la Tabla 1, que resume los porcentajes de aproximación de todos los modelos con las

referencias de respuesta real. La Tabla 1 es un cuadro comparativo de las respuestas de los modelos a las diferentes entradas y las señales reales de salida del sistema motor.

Tabla 1. Aproximaciones de los modelos

	Salida Modelo 01	Salida Modelo 02	Salida Modelo 03	Salida Modelo 04	Salida Modelo 05	Salida Modelo 06	Salida Modelo 07	Salida Modelo 08	Salida Modelo 09	Salida Modelo 10
Salida Real 01	81,01	80,3	66,28	34,52	35,39	73,24	49,25	44,3	-49,72	55,76
Salida Real 02	77,77	78,26	49,57	24,41	-107,1	64,05	14,94	-18,17	48,2	31,37
Salida Real 03	75,83	74,7	72,83	67,18	-33,21	77,43	40,42	13,18	9,334	64,07
Salida Real 04	43,02	39,53	52,39	74,88	63,76	54,68	33,65	56,78	-38,1	52,06
Salida Real 05	79,31	77,12	81,87	66,21	84,57	83,71	67,92	73,73	-36,16	79,18
Salida Real 06	79,71	79,02	80,33	74,45	-74,5	80,87	66,5	-48,53	24,81	78,41
Salida Real 07	78,48	78,12	78,66	66,93	32,92	78,26	78,75	46,93	-67,38	78,18
Salida Real 08	79,8	79,42	79,89	65,21	67,42	79,28	83,61	85,09	-79,38	79,37
Salida Real 09	81,23	80,2	77,19	70,19	-117,5	81,62	51,92	-36,06	72,96	69
Salida Real 10	78,44	76,47	84,96	75,47	-88,16	84,2	73,34	-55,93	-27,58	85,26

En la Tabla 1 podemos apreciar que el modelo 6 posee mejores aproximaciones con todas las señales de referencia. El objetivo es encontrar el mejor modelo que aproxime a la función de transferencia dado en (1).

$$G(s) = \frac{K_p}{T_{p1}T_{p2}s^2 + (T_{p1} + T_{p2})s + 1} \quad (1)$$

El modelo del motor DC para la posición angular del motor DC viene dado por:

El motor DC viene dado por:

$$\frac{\theta(s)}{V(s)} = \frac{K_t}{[Js + b](Ls + R) + K_t K_e} s \quad (2)$$

Donde, los parámetros son: inercia (J), coeficiente de fricción viscosa (b), Resistencia de armadura (R), inductancia de armadura (L), constante de motor (K_t), y constante contra electromotriz (K_e).

Podemos usar la aproximación $K_t = K_e$ en los motores DC para simplificar el número de parámetros.

Además, se requiere de un integrador y además multiplicar por el factor de ganancia K , de modo que tengamos la representación de la función de transferencia como voltaje de salida sobre voltaje entrada, dando la forma:

$$\frac{K\theta}{V} = \frac{K^2}{JLs^3 + (JR + bL)s^2 + (bR + K^2)s} \quad (3)$$

Comparando coeficientes las ecuaciones (1) y (3), establecemos las siguientes relaciones:

$$K^2 = K_p \quad (4)$$

$$JL = T_{p1}T_{p2} \quad (5)$$

$$JR + bL = T_{p1} + T_{p2} \quad (6)$$

$$bR + K^2 = 1 \quad (7)$$

En la Tabla 2 resume las características de los diferentes modelos que presentan las mejores aproximaciones.

Tabla 2. Parámetros de los mejores modelos obtenidos desde el IDENT de Matlab.

	Modelo 2	Modelo 3	Modelo 6	Modelo 7
K_p	0,76382	0,80089	0,79776	0,79816
T_{p1}	0,001	0,001	0,0061359	0,16354
T_{p2}	0,001	0,020535	0,0011009	0,001

Con estos valores proporcionados por la Tabla 2, podemos obtener los parámetros del modelo y junto con las ecuaciones dadas en (4), (5), (6), y

(7) obtenemos los coeficientes de los parámetros del motor DC que se listan en la siguiente Tabla 3.

Tabla 3. Parámetros del motor DC EMG30.

Símbolo	Descripción	Valor	Unidades
J	Momento de inercia equivalente.	0.0072	Kgm ²
b	Fricción viscosa equivalente.	0.0389	Nm rad/s
K _v =K	Constante de fuerza electromotriz.	0.893174115165	V/rad/s
K _t =K	Constante de torque de motor.	0.893174115165	Nm/A
R	Resistencia de armadura	5.2	Ω
L	Inductancia de armadura	9.381x10 ⁻⁴	H

El valor de la resistencia se realizó mediante medición directa entre los cables de alimentación de cada motor siendo $R = 5.2$. Con ello podemos conformar la función de transferencia del motor DC dado por:

$$\frac{\omega}{V} = \frac{0.893174115165}{6.75501231 \times 10^{-6}s^2 + 7.2368 \times 10^{-3}s + 1} \quad (8)(8)$$

CONTROLADOR DIFUSO PD+I

El uso de técnicas de control clásico como es el control PID de ganancia fija, en algunos casos resulta ser una buena alternativa para controlar sistemas dinámicos; ya que proporcionan tiempos de respuesta rápidos, sin embargo entre mayor es la precisión requerida en el sistema el ajuste de este tipo de control es más difícil ya que son bastante sensibles a las señales de ruido y en ocasiones introducen oscilaciones cuando se presentan retardos en el sistema. Cuando la dinámica de los sistemas o procesos a controlar es no lineal, el control tiene que tener la capacidad de compensar esta no-linealidad y aunque el control PID asume relaciones lineales, este no tiene la capacidad para responder a estas no linealidades. La no linealidad difícilmente puede ser caracterizada por una ecuación por lo que en la mayoría de los casos es tratada de manera subjetiva por el operador del proceso. Esta subjetividad tiene implicaciones profundas para poder modelar este tipo de sistemas a través de la lógica difusa, [3]. La implementación de controladores PD+I en hardware basados en lógica

difusa es motivada por su habilidad para capturar estrategias cualitativas de control y su capacidad de implementar un comportamiento de control altamente flexible. Con estos podemos lograr que nuestros sistemas puedan ajustarse a condiciones cambiantes que son muchas veces imposibles de predecir, tales como los cambios ambientales o las condiciones de desgaste en sus componentes físicos, por citar algunos ejemplos.

Dentro de los objetivos que nos planteamos resolver:

- Diseño de un Controlador Difuso PD+I para un Motor DC.
- Simular del Controlador Difuso PD+I en LabVIEW.
- Implementar en Tiempo Real del Controlador Difuso PD+I.

SINTONÍA DEL CONTROLADOR PID USANDO EL MÉTODO DE ZIEGLER NICHOLS

Para el diseño de un controlador PID mediante el método de Ziegler Nichols, se hace necesario conocer las características de respuesta de la planta (Motor DC EMG-30) a una entrada de referencia escalón unitario, la extracción de esas características se muestran en la Fig. 8.

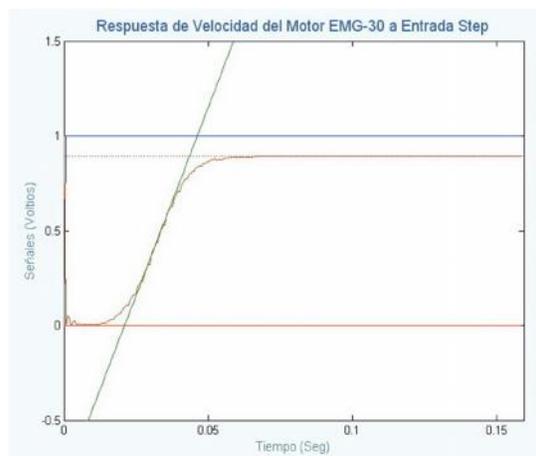


Fig. 8. Curva de reacción de Ziegler Nichols

Los parámetros resultantes desde la curva de reacción de la Fig. 8 son:

$$\begin{aligned}
 T &= 0.0225 \\
 K &= 0.8931 \\
 L &= 0.02105 \\
 a &= 0.8355
 \end{aligned}$$

De acuerdo a las tablas proporcionadas por Ziegler Nichols obtenemos los parámetros de sintonía del controlador PID que se muestran en la Tabla 3.

Tabla 3. Parámetros PID Ziegler Nichols.

Par	PID	Comis.
$\frac{K_p}{a}$	$2 \frac{L}{T_L}$	$0.5 \frac{L}{T_d}$
1.4362	0.0421	0.0105

Los parámetros encontrados corresponden a la estructura del controlador PID dado por:

$$u^n = K_p \cdot \left(e^n + \frac{1}{T_i} \cdot \sum_{j=1}^n e_j \cdot \frac{1}{T_s} + T_d \cdot \frac{e^n - e^{n-1}}{T_s} \right) \quad (9)$$

DISEÑO DEL LAZO DE CONTROL DIFUSO PD+I

Supongamos que se desea controlar un sistema SISO (de entrada y salida única). Resulta entonces conveniente utilizar como variables de entrada al controlador, el error existente en el proceso ($e = r - y$) el cambio del error (de/dt) y la acumulación del error ($\int edt$). Luego cada una debe ser fuzzyficada, es decir, se deben definir una cantidad de funciones de pertenencia que representen los diferentes estados (lingüísticos) que pueden tomar estas variables (por ejemplo, error pequeño, mediano o grande). Para realizar este punto se dispone de una gran cantidad de funciones, tales como gaussianas, triangulares, trapecios, etc.

El procedimiento para definir las funciones debe ser el siguiente:

- Se elige un número de funciones, de tal modo que se represente todos los estados que la variable tome.
- Se diseñan de un ancho mínimo suficiente que permita obviar el ruido de medición.

- Se desea un cierto grado de cruzamiento entre las distintas funciones, para no incurrir en estados pobremente definidos.
- Se recomienda partir con funciones simétricas, para posteriormente realizar ajustes.
- Inicialmente se recomienda un entrecruzamiento del 50%, de tal modo que cada elemento quede representado en al menos dos funciones.

Luego se deben definir las reglas base, cuya función es establecer una vinculación lógica entre los grados de pertenencia de las distintas variables de entrada, por ejemplo, si el error es grande y crece entonces se debe aplicar actuación correctiva grande. Se definen entonces todas las sentencias lógicas posibles, asignándole a cada una su actuación correspondiente. Luego, las actuaciones deben ser defuzzyficadas, para ser convertidas a número, lo que se logra mediante diversos métodos, siendo el más utilizado el del centroide, [2].

Una mayor cantidad de entradas, se tiene una mayor cantidad de reglas base, lo que dificulta crecientemente el diseño del controlador. Por ello, conviene retirar el factor integrativo de las entradas al controlador Fuzzy, para incorporarlo en forma individual sumándolo a la actuación final. Se tiene entonces una especie de controlador híbrido en que el error y el cambio del error son variables de tipo difuso, y la acumulación del error es determinística (*crisp*). El cambio anterior queda ilustrado en la Fig. 9.

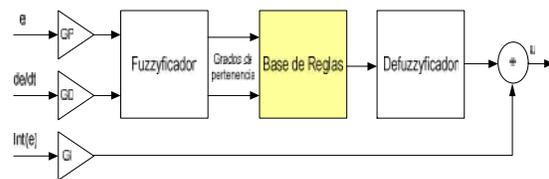


Fig. 9 Diagrama del controlador difuso con ganancias.

INTRODUCCIÓN A LAS GANANCIAS DE AJUSTE

El controlador difuso PD+I presenta tres términos de entrada: error, error integral, y el error

derivativo. Una base de la regla con tres entradas, fácilmente llega a ser bastante grande y, como se mencionó anteriormente, las reglas acerca de la acción integral son una molestia. Por consiguiente es común separar la acción integral como en el controlador difuso PD+I (DPD+I) que se muestra en la Fig. 10.

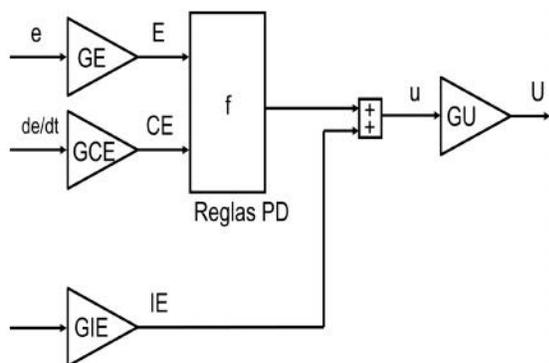


Fig. 10 Controlador difuso PD+I

La aproximación lineal para el controlador es:

$$u = GE + GU \cdot \left(e_n + \frac{GCE}{GE} \frac{de_n}{dt} + \frac{GIE}{GE} i e_n \right) \quad (10)$$

En la última línea asumimos a GE no cero. Comparando (9) y (10) las ganancias son relacionadas de la siguiente forma.

$$GE + GU = Kp \quad (11)$$

$$\frac{GCE}{GE} = \tau_d \quad (12)$$

$$\frac{GIE}{GE} = \frac{1}{\tau_i} \quad (13)$$

Siendo los valores numéricos dados en la Tabla 4.

Tabla 4. Parámetros del controlador Difuso PD+I.

4. Parámetros PD+I	GU	GCE	GIE
1	1.4362	23.7530	0.0105

ALGORITMO DE CONTROL DIFUSO

Considerando la descripción de la planta realizada y modelo de la misma, se procedió a diseñar un sistema en base a lógica difusa mediante el cual se pudiese controlar de buena forma la velocidad del motor DC. Para ello, se empleó un controlador que recibe como entradas el error e , definido como la diferencia entre el ángulo deseado y el medido; y la derivada del error de/dt . Este controlador entrega una salida proporcional a la rapidez de la tensión de actuación u sobre el motor DC. Para este diseño, se emplearon 7 conjuntos difusos para cada variable de entrada y 7 conjuntos difusos para la variable de salida, haciendo en total 49 reglas (ver Tabla 5).

Tabla 5. Base de reglas difusas.

Error e	Derivada del error $\frac{de}{dt}$							
	NB	NM	NS	ZO	PS	PM	PB	
NB	NB	NB	NB	NB	NM	NS	ZO	
NM	NB	NB	NB	NM	NS	ZO	PS	
NS	NB	NB	NM	NS	ZO	PS	PM	
ZO	NB	NM	NS	ZO	PS	PM	PB	
PS	NM	NS	ZO	PS	PM	PB	PB	
PM	NS	ZO	PS	PM	PB	PB	PB	
PB	PB	PB	PB	PB	PB	PB	PB	

La base de reglas tiene estructura IF – THEN, las funciones de pertenencia triangulares, implicancia mediante el operador MIN y defuzzyficacion mediante el método del centroide.

Para implementar este controlador, se utilizó el Toolkit de Lógica Difusa de LabVIEW, [5 y 6]. Los resultados de la simulación del controlador Difuso PD+I muestra la ventana interactiva en el panel frontal de la Fig. 11.

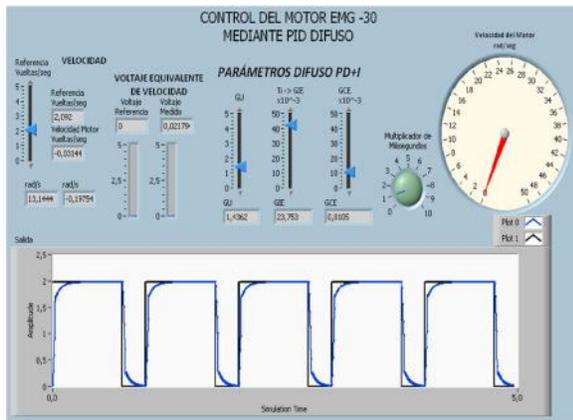


Fig. 10 Front panel del controlador difuso PD+I.

Mientras que el programa del controlador es mostrado en la Fig. 12.

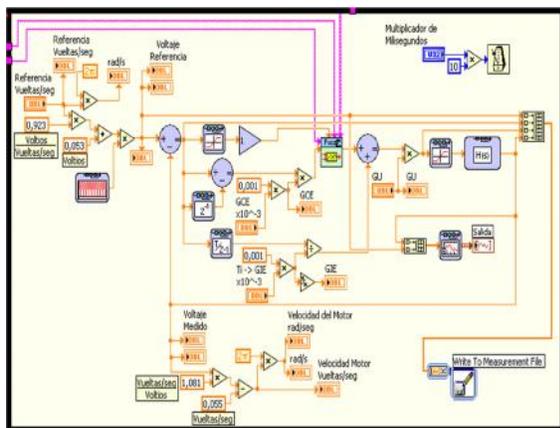


Fig. 11 Block diagram del controlador difuso PD+I.

IMPLEMENTACION EN TIEMPO REAL

La implementación real del sistema presenta los siguientes componentes físicos (hardware): la tarjeta de adquisición de datos, PCI6024E.

Computadora Pentium IV con los lenguajes de programación LabVIEW y Matlab, un osciloscopio, generador de funciones, fuente bipolar, y el motor DC EMG-30. Este sistema se muestra en la Fig. 12.

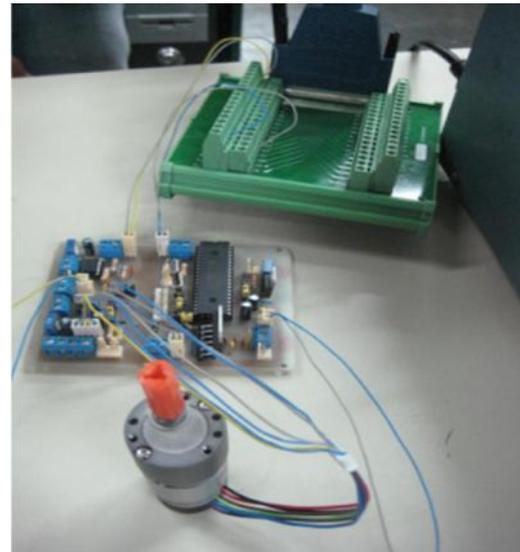


Fig. 12 Implementación del sistema real.

Para la realización del control de la velocidad del motor DC en tiempo real se utilizó el programa LabVIEW que es un ambiente de programación gráfico amigable para el usuario.

Tiene muchas características para trabajar con las tarjetas de adquisición de datos comerciales como la PCI6024E que tiene la capacidad de conversión A/D y D/A, entradas y salidas digitales (digital I/O) y contadores-timers (Counter/Timers). [4].

CIRCUITOS ELECTRÓNICOS.

Las limitaciones físicas de algunas tarjetas de adquisición de datos, implican simplificar las señales de los sensores a entradas analógicas, ya que es lo más común de realizar, para el caso del control de velocidad del motor además tenemos que procesar la señal de control y amplificar dicha señal adecuadamente a PWM, ya que es la forma más común de realizar el control de un motor DC.

A continuación se describen los circuitos electrónicos empleados y las características de los mismos.

Lectura de encoder

Este circuito se encarga de efectuar la correcta lectura del encoder que posee el motor EMG-30, cuyo diagrama esquemático (ver Fig. 13).

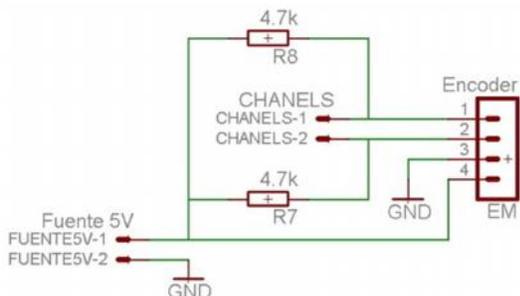


Fig. 13 Circuito para lectura de encoder.

Este circuito requiere de una Fuente de Voltaje de 5V y el tipo de conexión para los canales A y B del encoder es PullUp con resistencias de 4.7kΩ.

Convertidor frecuencia voltaje

La mayor cantidad de entradas análogas en una tarjeta de adquisición de datos, implica que la señal del encoder deba ser convertida, la forma más común de realizar dicha transformación es convirtiendo la frecuencia en voltaje en una relación lo más lineal posible, ya que la frecuencia de señal emitida por un encoder representa velocidad, se deben limitar adecuadamente los niveles de voltaje que corresponden a una frecuencia determinada o velocidad, para esto usaremos el integrado LM331 que posee una linealidad desde 1Hz hasta 10KHz, la configuración básica del circuito se muestra en la Fig. 14.

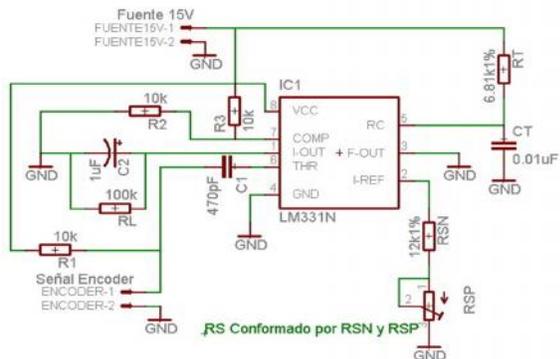


Fig. 14 Circuito para conversión de frecuencia a voltaje.

Señal de control en PWM

Luego de procesar la señal de error por parte de la PC es decir luego de efectuar la corrección del error, el envío de la señal de control se hace mediante uno de los puertos análogos de salida de la DAC PCI6024E, esta señal no posee las características suficientes como para ingresar directamente al motor sino que debe ser procesada adecuadamente; la forma de control más común y normal es mediante PWM, es decir, la conversión de la señal de voltaje de control en un señal de frecuencia constante con un ancho de pulso variable que depende de los niveles de voltaje, es decir toda la señal de control es transferida al motor pero con diferentes anchos de pulso y no hay pérdidas debido a que es a voltaje constante y lo único que varía son los anchos de pulso.

La conversión de la señal de control a PWM la realizamos usando un microprocesador PIC16F877A, ya que dicho integrado posee un modulo especial de conversión de una señal análoga cualquiera en PWM, modulo que se aprovecha para el control de motores, pero nuestra aplicación solo busca usar el modulo para la conversión de la Señal de control a PWM ya que el objetivo es diseñar controladores y probarlos desde la PC en Tiempo Real.

Las conexiones que se requieren para el funcionamiento del módulo PWM se muestran en la Fig. 15.

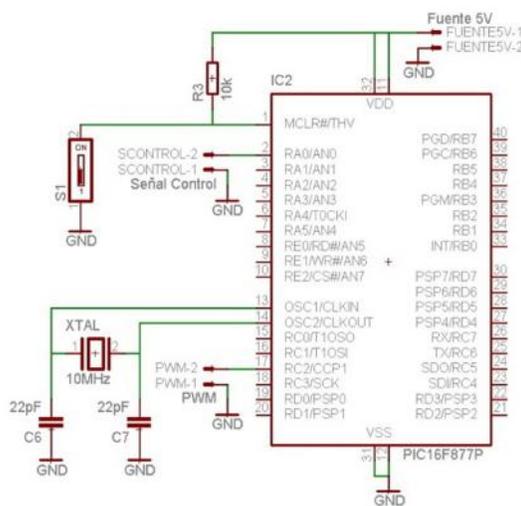


Fig. 15 Módulo PWM del PIC 16F877A.

Etapa de potencia

La etapa de potencia se hizo con el integrado L6203, que básicamente es un puente H especial para el control de motores y que soporta inclusive hasta 5A de corriente sin problema alguno, el esquemático de este circuito se presenta en la Fig. 16.

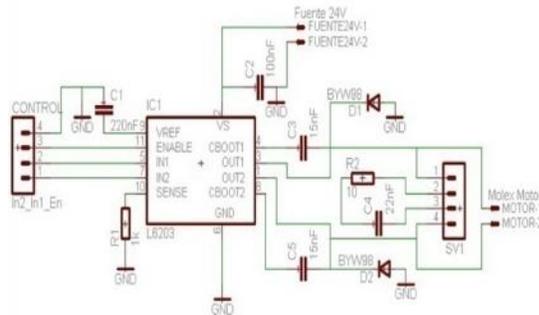


Fig. 16 Circuito amplificador de potencia.

La señal de PWM generada por el PIC 16F877A no posee las características de corriente necesarias para poder mover el motor a cualquier nivel de voltaje, probablemente solo se pueda generar velocidades pequeñas si solo usamos la señal directa del PIC, por ese motivo dicha señal debe ser amplificada para que los niveles de corriente que requiere el motor sean satisfechos.

Los resultados de respuesta del sistema en tiempo real se muestran en la Fig. 17.

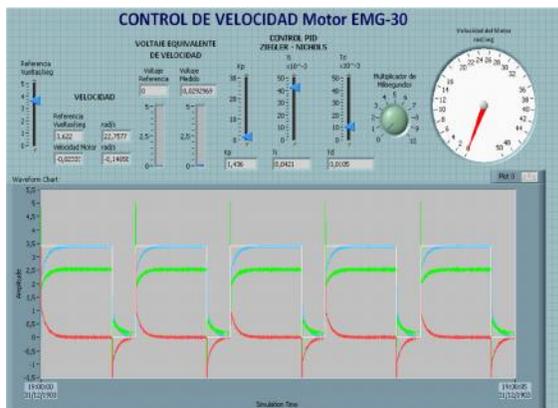


Fig. 17 Control de velocidad del motor DC en tiempo real.

La implementación del programa de control en tiempo real se muestra en la Fig. 18

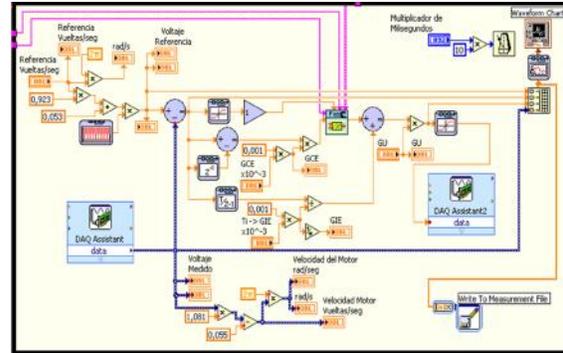


Fig. 18 Diagrama de adquisición de datos para el control de la velocidad del motor DC en tiempo real.

CONCLUSIONES

Para realizar este trabajo de investigación, se ha tenido que implementar el sistema para el control de la velocidad del motor DC. La implementación ha sido fundamental para manipular las variables físicas e interactuar convenientemente con la tarjeta de adquisición de datos PCI16024E.

El diseño del sistema de control se realizó mediante la técnica Difuso PD+I, en este caso se ha requerido del modelo para identificar los parámetros del motor DC, siendo estos resultados óptimos para el diseño del control.

La técnica de control Difuso PD+I cumple con las metas impuestas en este trabajo como son, mínimo sobrepaso, y error en estado estacionario nulo, los que se han verificado mediante la implementación en tiempo real y validando así la técnica de control desarrollada en este artículo.

REFERENCIAS

1. Jantzen, J., "Design Of Fuzzy Controllers". Technical University of Denmark. Denmark, 1998.
2. Jantzen, J., Tuning Of Fuzzy PID Controllers. Technical University of Denmark. Denmark, 1998.
3. Lotfi, Z., Fuzzy Logic Toolbox, User's Guide. Versión 2. The Math Works, Inc. 1995.

4. PCI6024E National Instruments. User Guide, 2005.
5. <http://www.ni.com/labview/esa/>
6. <http://www.ni.com/dataacquisition/esa/>

Correspondencia: robust@uni.edu.pe