

# DESARROLLO E IMPLEMENTACIÓN DE CONTROLADORES PARA LA PLANTA DE VELOCIDAD DC SPEED CONTROL SYSTEM MEDIANTE LA UTILIZACIÓN DE LA HERRAMIENTA RTW (REAL-TIME WORKSHOP) DE MATLAB

Roberto Javier González Puma  
César Andrés Villacís Fiallos  
Facultad de Ingeniería Electrónica, Escuela Politécnica del Ejército  
Av. El Progreso S-N, Sangolquí, Ecuador  
anvi\_2587@hotmail.com  
rj\_barsa@hotmail.com

**Resumen.-** El presente documento engloba el diseño e implementación de controladores: PI, PID, Digital e Inteligentes, para controlar la velocidad de la unidad DC SPEED CONTROL SYSTEM, perteneciente al laboratorio de Control Industrial del Departamento de Eléctrica y Electrónica de la Escuela Politécnica del Ejército. Se utilizó de la herramienta RTW (Real-Time Workshop) de Matlab que nos permite ejecutar en tiempo real los controladores diseñados. La adquisición de datos de la planta se lo realiza con la tarjeta NI PCI 6221. El artículo brinda información sobre el diseño e implementación de los controladores mencionados en un sistema real de velocidad. Los modelos de simulación y los modelos experimentales se desarrollan en la herramienta Simulink de Matlab.

**Palabras Clave.-** DC SPEED CONTROL SYSTEM, optimización, ubicación de polos, ecuación diofantina, lógica difusa, control neuronal, control predictivo.

## I. INTRODUCCIÓN

El diseño e implementación de controladores para procesos industriales en la actualidad es una de las especialidades relevantes en la Ingeniería de Control Automático.

Con el propósito de obtener un sistema didáctico que permita a los estudiantes conocer paso a paso el diseño e implementación de controladores para la Unidad DC Speed Control System por medio de la herramienta Real Time Workshop de Matlab se

diseñó un modelo matemático que simula la unidad DC Speed Control System.

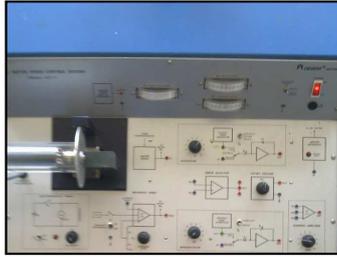
Los controladores implementados para la unidad, fueron: PI, PID, Digital, PI FUZZY, Predictivo y Neuronal por modelo de referencia.

Se realizó experimentos en tiempo real, con la ayuda de la herramienta Real Time Workshop de Matlab para acoplar los controladores diseñados a la unidad de velocidad DC Speed Control System, donde se observó las variables de error, control y salida del sistema controlado, producidas al variar la entrada del sistema.

## II. CARACTERISTICAS DEL SISTEMA A IMPLEMENTAR

### CARACTERÍSTICAS DE LA UNIDAD DE VELOCIDAD DC SPEED CONTROL SYSTEM.

En la figura 1 se puede observar la unidad de velocidad DC Speed Control System la cual posee los siguientes módulos: interfaz de motor, medidor M2, freno electromagnético, entre otros.



**1Figura 1 Unidad de velocidad DC SPEED CONTROL SYSTEM**

La interfaz de motor, sirve para energizar al motor DC. Maxon cuyas características se presentan la figura 2.

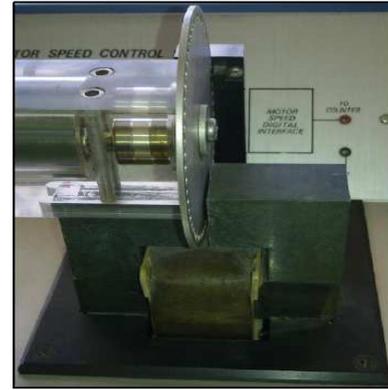


**Figura 2. Características del Motor**

Para conocer los valores de velocidad que genera el motor DC se acopla un tacómetro que convierte la energía rotacional producida por el eje del motor en energía eléctrica.

El Medidor M2, es un galvanómetro que mide la señal de voltaje DC generada por el tacómetro, el cual permite al usuario observar las variaciones de voltaje en un rango de -4000 a 4000 RPM.

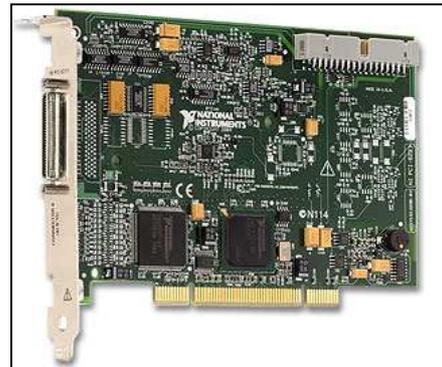
El Freno Electromagnético es un dispositivo electromagnético que actúa directamente sobre un disco de aluminio acoplado al eje del motor demostrado en la figura 3.



**Figura 3. Acoplamiento disco-eje del motor.**

### **CARACTERÍSTICAS DE LA TARJETA DE ADQUISICIÓN DE DATOS NATIONAL INSTRUMENT PCI-6221.**

La PCI-6221 de National Instruments (figura 4) es una tarjeta de adquisición de datos (DAQ) multifunción de bajo costo con un formato físico tipo PCI, permitiéndole funcionar en diferentes sistemas operativos como Windows, Linux, Mac, entre otros.



**Figura 4: Tarjeta DAQ PCI-6221**

Con este tipo de tarjeta de adquisición de datos, podemos enviar desde el computador y recibir desde el exterior la señal producida por sensores y transductores; este tipo de señales tanto de entrada como de salida son del tipo digital, analógico, frecuencia.

Para conectar las entradas y salidas de la tarjeta DAQ PCI-6221 se lo realiza mediante un cable NI SHC68-68.EPM (figura 5), el cual posee cables cruzados blindados individuales para las entradas y salidas analógicas y digitales.



Figura 5: Cable NI SHC68-68.EPM

Como elemento final de conexión de las señales se tiene un conector de tipo SCB-68 (figura 6), este conector, posee 68 pines de montaje vertical ajustados por tornillos para su fácil utilización. La distribución que posee el conector de tipo SCB-68 se lo muestra en la figura 7.



Figura 6: Conector SCB-68

AI 0	68	34	AI 6
AI GND	67	33	AI 1
AI 9	66	32	AI GND
AI 2	65	31	AI 10
AI GND	64	30	AI 3
AI 11	63	29	AI GND
AI SENSE	62	28	AI 4
AI 12	61	27	AI GND
AI 5	60	26	AI 13
AI GND	59	25	AI 6
AI 14	58	24	AI GND
AI 7	57	23	AI 15
AI GND	56	22	AO 0
AO GND	55	21	AO 1
AO GND	54	20	NC
D GND	53	19	P0.4
P0.0	52	18	D GND
P0.5	51	17	P0.1
D GND	50	16	P0.6
P0.2	49	15	D GND
P0.7	48	14	+5 V
P0.3	47	13	D GND
PFI 1/P2.3	46	12	D GND
PFI 10/P2.2	45	11	PFI 0/P1.0
D GND	44	10	PFI 1/P1.1
PFI 2/P1.2	43	9	D GND
PFI 3/P1.3	42	8	+5 V
PFI 4/P1.4	41	7	D GND
PFI 13/P2.5	40	6	PFI 5/P1.5
PFI 15/P2.7	39	5	PFI 6/P1.6
PFI 7/P1.7	38	4	D GND
PFI 8/P2.0	37	3	PFI 9/P2.1
D GND	36	2	PFI 12/P2.4
D GND	35	1	PFI 14/P2.6

Figura 7: Distribución de pines para el conector SCB-68

### III. IDENTIFICACIÓN DE LA UNIDAD DC SPEED CONTROL SYSTEM

Para la identificación de un sistema se tienen que seguir los siguientes pasos: recolección de datos de entrada y salida, selección de un modelo apropiado, criterio de ajuste y validación del modelo.

La recolección de datos de entrada y salida, se realizó por medio de la tarjeta de adquisición de datos DAQ PCI-6221 utilizando la entrada analógica AI0 (borna 68), salida analógica AO0 (borna 22) y tierra GND (borna 56), dichas conexiones se las realiza como se observa en la figura 9.

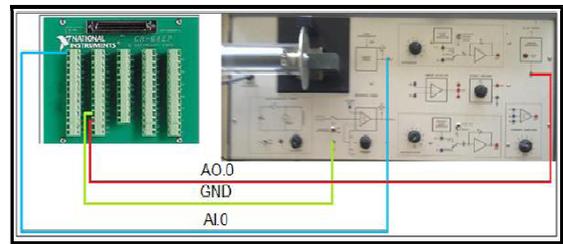


Figura 9: Diagrama de conexiones tarjeta unidad de velocidad.

Después de realizar las conexiones pertinentes entre el modulo de velocidad y la tarjeta de adquisición de datos, se realiza un programa en Simulink que nos permite observar en la pantalla del computador todas las señales de interés que se van generando en el transcurso de los experimentos.

En la figura 10.1 y 10.2, se observa la configuración del tiempo de muestreo, canal de salida y rango de salida para la entrada y salida analógica de la tarjeta de adquisición de datos PCI-6221.

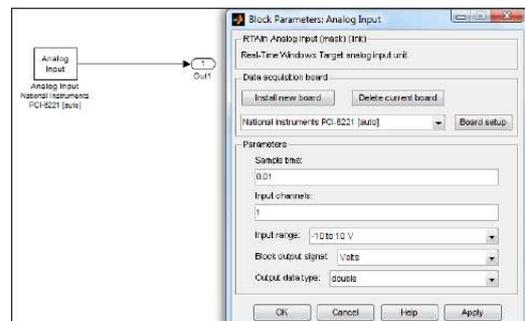
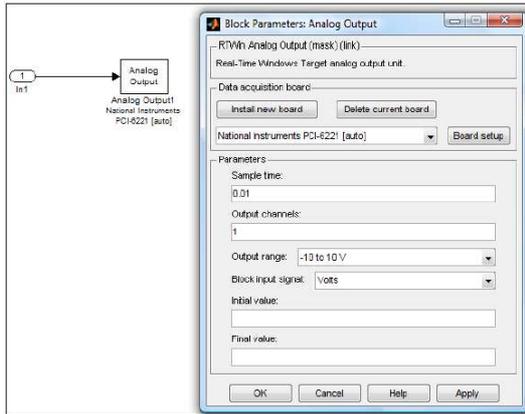
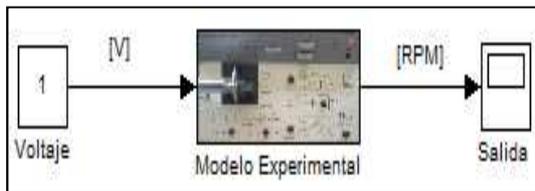


Figura 10.1 Configuración de entrada analógica



**Figura 10.2 Configuración de salida analógica**

El modelo para realizar la identificación del sistema, es una estructura del tipo caja negra (figura 11), en la cual se tiene una entrada (Voltaje) generando una salida (Velocidad en RPM). [2]



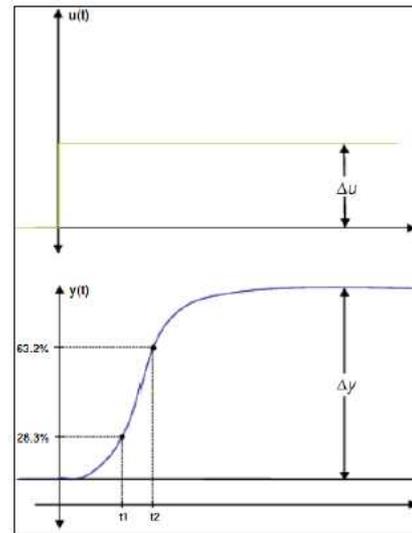
**Figura 11: Estructura tipo caja negra**

Dentro de la caja negra, se tiene un modelo de primer orden con tiempo muerto del que se necesita conocer los valores de la ganancia del sistema (K), constante de tiempo ( $\tau$ ) y tiempo muerto ( $\theta$ ). Todas estas constantes son obtenidas al observar la respuesta que tiene el sistema ante diferentes niveles de voltaje.

Una de las formas para obtener estas constantes es por medio del método de identificación de Smith. Este método estudia las respuestas temporales que tiene el sistema al 63.2% y 28.3% de su salida ante un cambio en su entrada (figura 11). Estas respuestas temporales determinan dos instantes de tiempo  $t_1$  y  $t_2$ ; con estos dos instantes y las ecuaciones 1 y 2 se obtienen las constantes  $\tau$  y  $\theta$ .

$$\theta + \frac{\tau}{3} = t_1 \quad (\text{ec 1})$$

$$\theta + \tau = t_2 \quad (\text{ec 2})$$

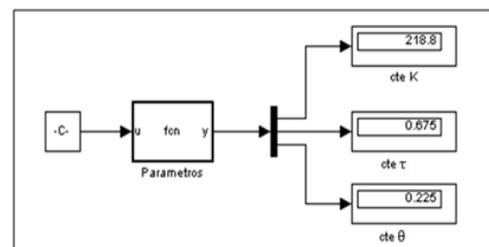


**Figura 12 Entrada escalón y puntos de estudio tomados de la salida del sistema.**

La ganancia de la planta K se obtiene de la relación de salida/ entrada de la planta  $\Delta y/\Delta u$ .

Por medio de Matlab, se determina los instantes de tiempo del sistema al 63.2% y 28.3% de su salida.

Con lo cual se calcula la ganancia del sistema (K), constante de tiempo ( $\tau$ ) y tiempo muerto ( $\theta$ ) (figura 13).



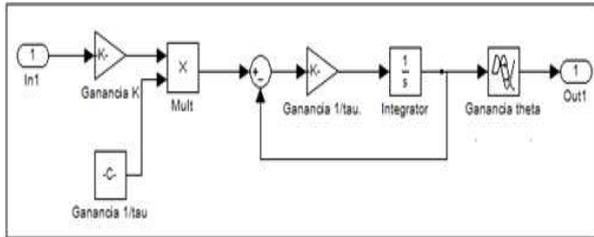
**Figura 13: Función para encontrar los valores K,  $\tau$  y  $\theta$ .**

Para obtener la respuesta más aproximada, se realizó experimentos a diferentes niveles de voltaje obteniendo los valores de la tabla 1.

Voltaje de entrada [V]	Velocidad de Salida [RPM]	Ganancia K	Tiempo muerto [s] $\theta$	Constante de Tiempo [s] $\tau$
1.4	0	0	0	0
2.5	225	131.5	0.285	0.615
3	500	166.28	0.26	0.6
4	825	206.23	0.22	0.63
5	1109.1	221.81	0.25	0.69
6	1431	238.5	0.24	0.69
7	1789.6	255.65	0.235	0.705
8	2075.8	259.5	0.26	0.675
9	2420.5	269	0.28	0.71

**Tabla 1: Valores de constantes y ganancias obtenidos a diferentes niveles de voltaje**

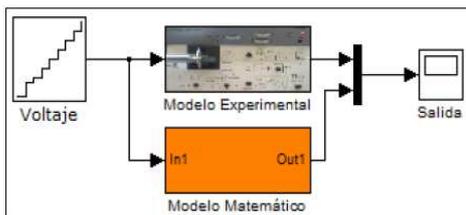
Promediando los valores obtenidos de las constantes  $K$ ,  $\tau$  y  $\theta$  se obtiene el sistema que se observa en la figura 14



**Figura 14: Diagrama del sistema de primer orden con tiempo muerto**

Como último paso en la identificación del sistema, se realizó la validación del modelo obtenido, sometiendo al modelo y a la planta de velocidad a una misma entrada de voltaje.

En la figura 15 se tiene la forma del diagrama en Simulink para realizar la validación.



**Figura 15: Diagrama en Simulink para validar el modelo**

#### IV. IMPLEMENTACION DE CONTROLADORES PI-PID

##### MARCO TEORICO

El objetivo de la optimización global es encontrar la mejor solución de modelos de decisiones difíciles frente a las múltiples soluciones locales [1].

Para modelos pequeños o medianos resulta adecuado el método simplex, el cual es un procedimiento iterativo que permite ir mejorando la solución a cada paso. El proceso concluye cuando no es posible seguir mejorando más dicha solución.

Matlab cuenta con la función `fminsearch`, que utilizando el método Simplex de optimización que logra encontrar la solución a la función objetivo, en principio es adecuado para funciones que puedan presentar discontinuidades. La sintaxis que presenta la función `fminsearch` es la siguiente:

`[x,fval,exitflag,output]=fminsearch(fun,x0,opciones)`

Donde:

`fun` = función objetivo a optimizar. Puede definirse en un M-archivo.

`x0` = Semilla del algoritmo. Puede ser un escalar, un vector o una matriz.

`opciones` = opciones de optimización donde se puede graficas o ver la evolución de las variables de interés.

`x` = mínimo local de `fun`.

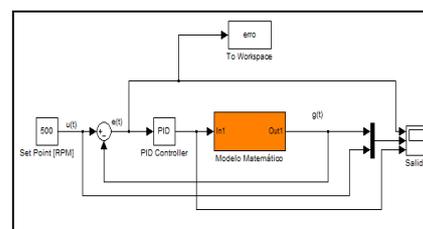
`fval` = devuelve el valor de la función objetivo `fun` en la solución `x`.

`exitflag` = describe la condición de salida de `fminsearch`.

`output` = Devuelve una estructura que contiene la optimización efectuada.

##### IMPLEMENTACION

Para encontrar las constantes del controlador PI y PID por medio de un método de optimización, se diseñó un lanzo de control (figura 16) en el que está involucrado el bloque del controlador PID y modelo de matemático de la planta de velocidad.



**Figura 16: Diagrama en simulink donde se envía la señal de error al espacio de trabajo.**

Dentro de las configuraciones para el controlador PID, se colocan las variables  $K_p$ ,  $K_i$  y  $K_d$  que son los parámetros de diseño para realizar la optimización.

La variable de interés para la función objetivo es el error que se produce entre el set point y la salida del sistema, por esta razón se lo envía al espacio de trabajo para ser utilizado.

Se genera un código en Matlab con el que se utiliza la función `fminsearch` con su función objetivo, que contiene la variable de interés error para el cálculo de los valores óptimos de  $K_p$  y  $K_i$ . Al ejecutar el código de Matlab se generan las variaciones que van tomando los valores iniciales, además se genera una gráfica que representa la evolución de la función objetivo con respecto a las iteraciones (figura 17 y 18).

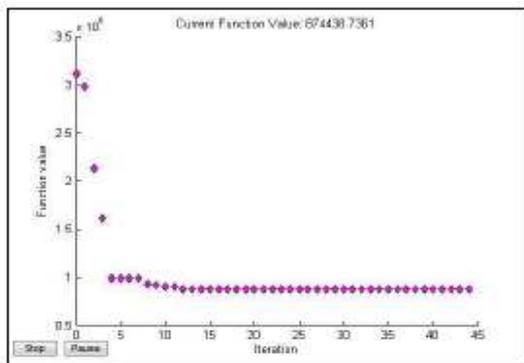


Figura 17: Gráfica de función objetivo vs iteraciones para controlador PI.

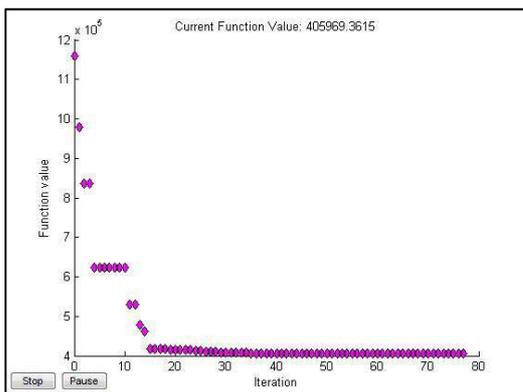


Figura 18: Gráfica de función objetivo vs iteraciones para controlador PID.

Para implementar los controladores PI y PID, utilizamos la tarjeta de adquisición de datos para

obtener la respuesta que tiene la planta de velocidad ante los diferentes niveles de referencia (Figura 19 y 20)

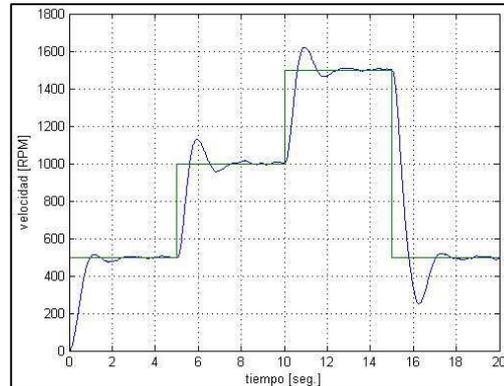


Figura 19: Señal de salida para el sistema controlado por un controlador PI

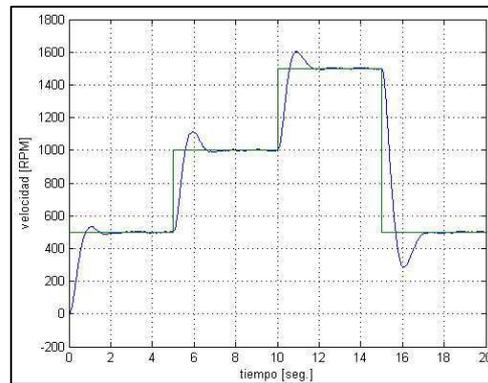


Figura 19: Señal de salida para el sistema controlado por un controlador PID.

## V. IMPLEMENTACION DEL CONTROLADOR DIGITAL

### MARCO TEORICO

En muchos casos las características de desempeño del sistema de control se especifican en términos de la respuesta transitoria para una entrada de escalón unitario, puesto que esta es fácil de generar y es suficientemente drástica.

La respuesta transitoria de un sistema de control puede representar con frecuencia oscilaciones amortiguadas antes de alcanzar el estado estacionario donde es común identificar:

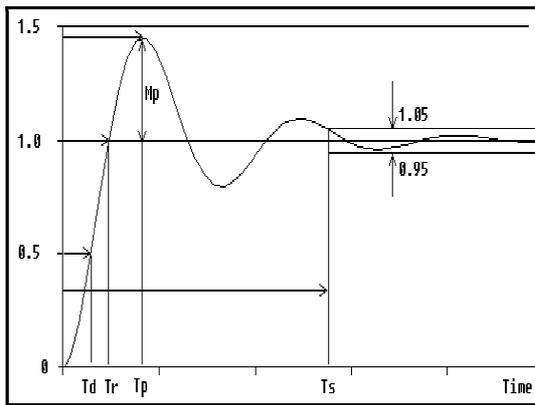
Tiempo de subida  $t_r$

Sobreelongación,  $M_p$

Tiempo de establecimiento,  $t_s$

Tiempo de retardo  $t_d$

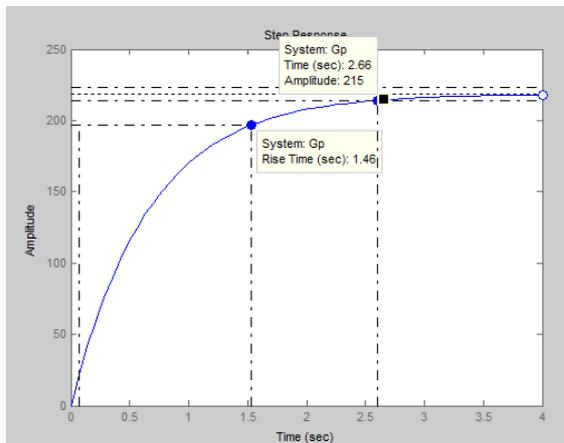
Las especificaciones en el dominio del tiempo que se han proporcionado y que presenta en la figura 20, son muy importantes ya que casi todos los sistemas de control son sistemas en el dominio del tiempo, es decir deben representar respuestas de tiempo estables. Lo que significa que el sistema de control debe modificarse hasta que la respuesta transitoria sea satisfactoria.



**Figura 20** Parámetros De Respuesta Transitoria Frente

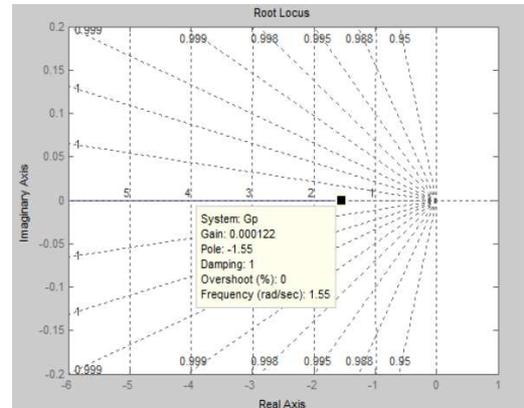
**IMPLEMENTACION**

Para el diseño del controlador digital se procede con el análisis de la respuesta transitoria, el cual permite conocer la respuesta del sistema frente a una entrada de tipo escalón como se muestra en la figura 21.



**Figura 21** Respuesta del sistema

Con la función “rlocus”, de Matlab se puede conocer la ubicación del polo como se muestra en la figura 22.



**Figura 22.** Polo del sistema

Estos resultados del sistema permiten establecer parámetros como el tiempo de subida, tiempo de establecimiento y la sobreelongación. Parámetros que permiten determinar las especificaciones del diseño con los cuales se mejorara la respuesta del sistema.

*tiempo de establecimiento*

$$t_s = 2.6 \text{ s}$$

*tiempo de subida*

$$t_r = 1.46 \text{ s}$$

La relación que permite establecer el diseño de la ubicación de polos con los parámetros de la respuesta transitoria se presenta a continuación:

$\epsilon$ : *factor de amortiguamiento*

$$M_p = e^{\frac{\epsilon\pi}{\sqrt{1-\epsilon^2}}} \times 100\%$$

$\omega_n$ : *frecuencia natural no amortiguada*

$$t_s = \frac{4}{\epsilon\omega_n}$$

Relaciones que permiten encontrar los nuevos polos de nuestro sistema tomando en cuenta que la variación del factor de amortiguamiento y de la frecuencia natural no amortiguada determina la

ubicación del polo como se presenta en la figura 23.

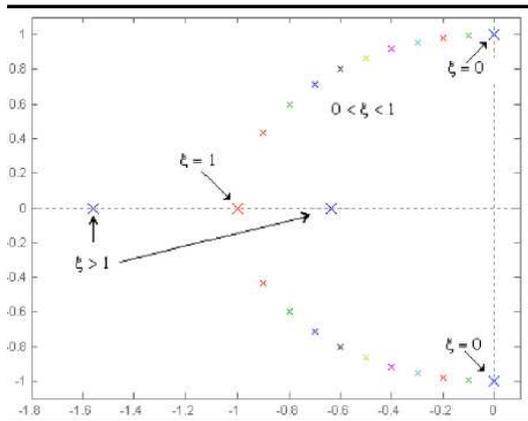


Figura 23: Ubicación de polos variando  $\varepsilon$

### Especificaciones del sistema

Las especificaciones para el diseño son:

$$\begin{aligned} Mp &\geq 5\% \\ ts &= 1.5s \text{ tiempo actual} \\ ess &= 0 \end{aligned}$$

Desarrollo:

$$\begin{aligned} Mp &= 5\% \\ Mp &= e^{\frac{-\pi\xi}{\sqrt{1-\xi^2}}} \times 100\% \\ \ln(0.05) &= \frac{-\pi\xi}{\sqrt{1-\xi^2}} \\ 2.90 &= \frac{\pi\xi}{\sqrt{1-\xi^2}} \\ 8.41 &= \xi^2(\pi^2 + 8.41) \\ \xi &= 0.67 \\ ts &= \frac{4}{\xi\omega_n} = 1.5 \\ \omega_n &= \frac{4}{0.67 * 1.5} \\ \omega_n &= 3.9 \text{ rad/s} \end{aligned}$$

Los nuevos parámetros de diseño establecen una nueva ubicación de los polos figura 24, los cuales se utilizan en el análisis del enfoque polinomial para encontrar el controlador deseado.

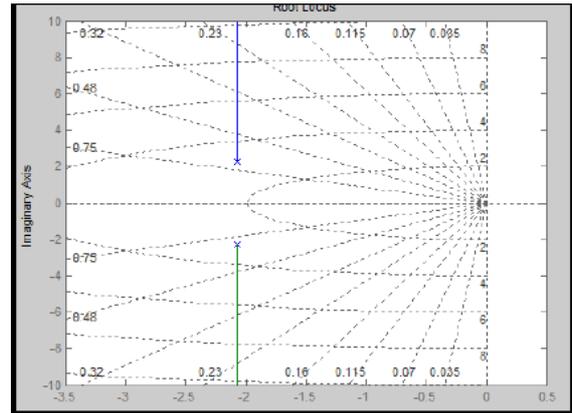


Figura 24: polos para diseño

La Ecuación Diofantina permite realizar un análisis diferente del diseño de controladores, donde se presenta un enfoque de ecuaciones polinomiales para el diseño de sistemas de control. Construyendo matrices que permiten encontrar nuestro controlador.

$$E = \begin{bmatrix} 0 & 1 \\ 0.3267 & -0.9857 \end{bmatrix}$$

Matriz de parámetros de la función de transferencia.

$$D = \begin{bmatrix} d_2 \\ d_1 + 0.9857 \end{bmatrix}$$

Matriz de valores de de polos y parámetros de la función de transferencia.

$$M = \begin{bmatrix} \beta_1 \\ \alpha_1 \end{bmatrix}$$

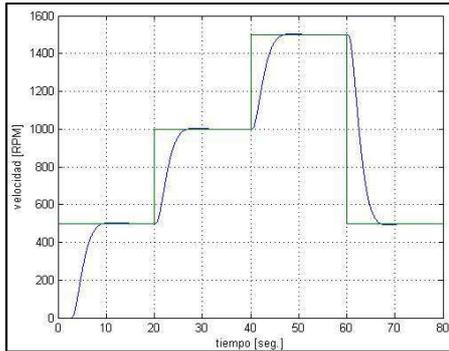
Matriz de parámetros del controlador

Resolviendo para M:

$$M = E^{-1} D$$

$$M = (\text{inv}(E)) * D$$

Para implementar el controlador digital utilizamos la tarjeta de adquisición de datos para obtener la respuesta que tiene la planta de velocidad ante los diferentes niveles de referencia (Figura 25).



**Figura 25: Señal de salida para el sistema controlado por un controlador digital**

## VI. IMPLEMENTACION DE CONTROLADOR PREDICTIVO

### MARCO TEORICO

El control predictivo tiene como objetivo resolver de forma efectiva, problemas de control y automatización de procesos industriales que se caractericen por presentar un comportamiento dinámico complicado, multivariable, y/o inestable. La estrategia de control en que se basa este tipo de control, utiliza el modelo matemático del proceso a controlar para predecir el comportamiento futuro de dicho sistema, y en base a este comportamiento futuro puede predecir la señal de control futura.

El Control Predictivo Basado en Modelo (CPBM), conocido también como Model Based Predictive Control (MBPC) o simplemente Model Predictive Control (MPC). Esta estrategia también se conoce como control por horizonte deslizante, por ser ésta la forma en la que se aplican las señales de actuación.

En Matlab existe la herramienta MPCTool que permite realizar diferentes tipos de controladores predictivos que pueden ser adaptados a la necesidad del usuario.

### IMPLEMENTACION

Para el diseño del controlador predictivo se utiliza la herramienta mpctool de Matlab, en la cual es necesario utilizar una función de transferencia lineal del sistema la cual es identificada por medio de la herramienta ident de Matlab.

La función de transferencia que se utiliza es la siguiente:

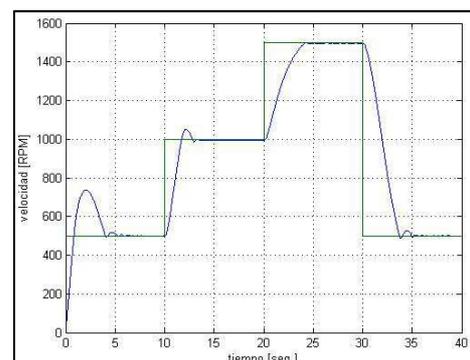
$$H(s) = \frac{4004}{s^2 + 10.51s + 15.02}$$

Conocida ya la función de transferencia del sistema, se establecen los valores de funcionamiento de la planta y los parámetros de diseño para el control predictivo, estos parámetros y valores están especificados en la tabla 2.

Parámetro	Detalle	Valor
Modelo y Horizonte	Intervalo de Control	0.1
	Horizonte de Predicción	10
	Horizonte de Control	4
Restricciones	Entrada : Voltaje	Min: 3 Max: 6.5
	Salida: Velocidad	Min: 500 Max: 1500
Pesos	Overall	0.6
	Varianza del peso	0.1
	Velocidad	1

**Tabla2: Parámetros de diseño para el controlador predictivo.**

Para realizar la implementación con el módulo de velocidad, se enlaza el controlador predictivo al modelo experimental, del cual se obtienen las señales del sistema controlado (figura 26).



**Figura 26: Señal de salida para el sistema controlado por un controlador predictivo.**

## VII. IMPLEMENTACION DEL CONTROLADOR PI FUZZY

### MARCO TEORICO

La lógica fuzzy se presenta como una técnica de la Inteligencia Artificial cuyo objetivo es expresar el conocimiento común, el cual es de tipo lingüístico cualitativo, en un lenguaje matemático regido por la teoría de conjuntos difusos y funciones de pertenencia asociados a estos. El control difuso está compuesto por la fuzzificación, Motor de Inferencia y la defuzzificación.

Conviene señalar que la aplicación de la lógica difusa en el control de procesos puede combinarse con otras técnicas de control convencionales, formando estructuras de control muy diversas como el controlador de tipo PI Fuzzy.

### IMPLEMENTACION

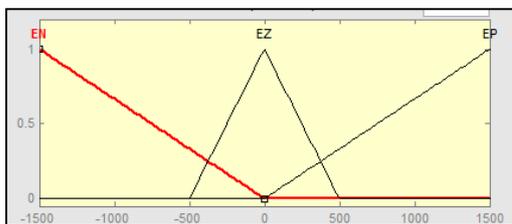
Para el diseño del controlador PI Fuzzy, se deben establecer las siguientes variables o conjuntos difusos: error, salida de voltaje e integral del error.

Para el error se establece un conjunto que va desde -1500 a 1500 RPM.

Para la integral del error se establece un conjunto que va desde los -10.000 a 10.000 RPM.

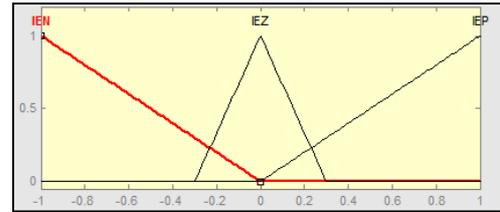
Finalmente, la salida de voltaje se establece en un rango de -7 a 7 voltios.

En el caso del error (figura 27), los conjuntos de pertenencia son designados como: error negativo (EN), error en cero (EZ) y error positivo (EP).



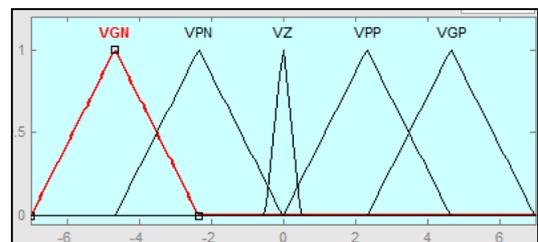
**Figura 27: Conjuntos de pertenencia para el error.**

Para la integral del error (figura 28), los conjuntos de pertenencia son designados como: integral del error negativo (EN), integral de error en cero (EZ) e integral del error positivo (EP).



**Figura 28: Conjuntos de pertenencia para la integral del error.**

En el caso de la salida de voltaje, se establecen 5 conjuntos de pertenencia (figura 29), los cuales son designados de la siguiente manera: voltaje grande negativo (VGN), voltaje pequeño negativo (VPN), voltaje en cero (VZ), voltaje pequeño positivo (VPP) y voltaje grande positivo (VGP).



**Figura 29: Conjuntos de pertenencia para la salida de voltaje.**

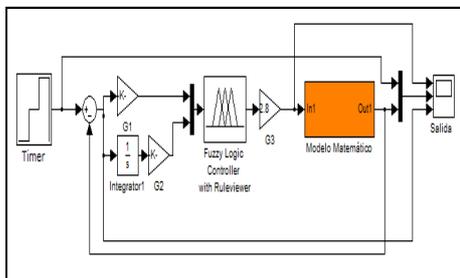
Luego de establecer los conjuntos de pertenencia para cada una de las variables, se realiza las combinaciones necesarias para formar la base de reglas. La base de reglas que se utiliza está basada en función de la máquina de inferencias de Mandami.

En la tabla 3, podemos observar las diferentes combinaciones de las variables de error e integral de error con su respectivo efecto que es la variable salida de voltaje.

Reglas	
1	si(error es negativo y cambio de error es negativo)entonces salida grande negativo
2	si(error es negativo y cambio de error es cero)entonces salida pequeño negativo
3	si(error es negativo y cambio de error es positivo)entonces salida cero
4	si(error es cero y cambio de error es negativo)entonces salida pequeño negativo
5	si(error es cero y cambio de error es cero)entonces salida cero
6	si(error es cero y cambio de error es positivo)entonces salida pequeño positivo
7	si(error es positivo y cambio de error es negativo)entonces salida cero
8	si(error es positivo y cambio de error es cero)entonces salida pequeño positivo
9	si(error es positivo y cambio de error es positivo)entonces salida grande positivo

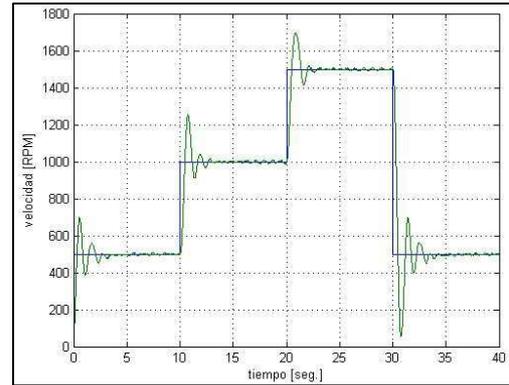
**Tabla 3: Base de reglas para el controlador PI Fuzzy**

Como éste controlador posee constantes que necesitan ser sintonizadas para ayudar a realizar el control, se realiza un programa en Matlab el cual resuelve un problema de optimización para encontrar las constantes  $K_p$  y  $K_i$  y  $G_3$  de controlador PI Fuzzy Figura 30.



**Figura 30. Sistema con controlador PI Fuzzy**

Para realizar la implementación con el módulo de velocidad, se enlaza el controlador PI FUZZY al modelo experimental, del cual se obtienen las señales del sistema controlado (figura 31).



**Figura 31: Señal de salida para el sistema controlado por un controlador PI FUZZY.**

## VIII. IMPLEMENTACION DEL CONTROLADOR NEURONAL

### MARCO TEORICO

El control de sistemas por modelo de referencia consiste en diseñar un sistema que modifique el comportamiento natural de la planta con el objetivo de aproximar la respuesta que tiene un modelo de referencia establecido. En este esquema de control el diseñador tiene el suficiente conocimiento del sistema como para definir el comportamiento deseado por medio de un modelo. Uno de los primeros trabajos de la aplicación de las redes neuronales para el control por modelo de referencia de sistemas puede consultarse en (K. S. Narendra, K. Parthasarathy, 1990).

### IMPLEMENTACION

Para realizar el controlador neuronal, se identifica la unidad de velocidad por medio de una red neuronal, usando un conjunto de valores que realizan la identificación de la misma, se define una variable denominada estado la cual está comprendida entre 240 a 1573 rpm y una variable denominada entrada la cual va de 2 a 7 voltios.

```

entrada=2:7;
estado=240:100:1573;
Pm=combvec(estado,entrada);
timestep=5;
Tm=[];
length(Pm)

```

Los patrones de entrenamiento están definidos en la matriz Pm mientras que la matriz de objetivos está definida por la matriz Tm que representan los valores de entrada y salida de la red neuronal identificada.

```
for i=1:length(Pm)
    Vin=Pm(2,i);
    Vel_in=Pm(1,i);
    sim('Velocidad_Modelo',[0 timestep]);
    dVel=Vel_out-Vel_in
    Tm=[Tm [dVel]];
end
```

Los valores de Tm son obtenidos al simular las diferentes combinaciones que tiene Pm,

Para la determinación de la red neuronal de control se lo realiza por medio de un modelo de primer orden del tipo:

$$G(s) = \frac{1}{1 + \tau s}$$

Que imita al modelo de la planta de velocidad, después de varias pruebas se encontró de que valor de  $\tau$  es de 1.4 seg.

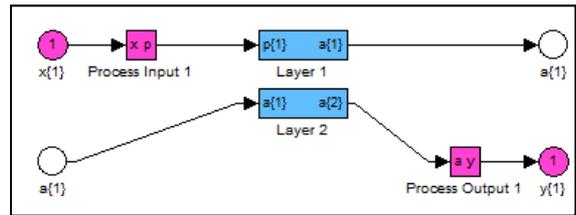
Para la red neuronal de control se evalúa la respuesta de la planta de primer orden, con respecto a los valores obtenidos de Pc, para posteriormente encontrar los valores de Tc.

```
entrada=2:7;
estado=240:100:1573;
% Matriz Pc y Tc
Pc=combvec(estado,entrada);%combinación de
los vectores de estados iniciales
%matriz de estados iniciales "Pc"
timestep = 10;
Q = length(Pc);
Tc = []; %matriz de objetivos "Tc".
for i=1:Q
    Vin = Pc(2,i);
    Vel_in = Pc(1,i);
    %Modelo para el aprendizaje
    sim('Velocidad_Modelo_Lineal',[0
timestep]);
    dVel = Vel_out-Vel_in;
    Tc=[Tc [dVel]];
end
```

Al conocer ya los valores de Pc y Tc, se crea una red neuronal con 8 neuronas en la capa oculta las cuales son de tipo tansig y purelin

```
S1 = 8;
cnet = newff(minmax(Pc),[S1 1],{'tansig'
'purelin'});
```

La red total de control viene a ser dada por la red neuronal identificada mnet y la red de control cnet, distribuidas de forma que dos capas sean usadas para la identificación mnet y dos capas para el control cnet (figura 32).

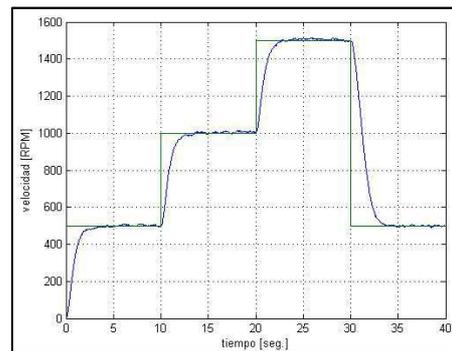


**Figura32: Asignación de la red neuronal total de control.**

Finalmente se hace la asignación y aprendizaje de los pesos y bias de las entradas de la red neuronal de control cnet a red total de control tnet

```
cnet.IW{1,1}(:,1) = tnet.IW{1,1};
cnet.IW{1,1}(:,2) = tnet.IW{1,2};
cnet.b{1} = tnet.b{1};
cnet.b{2} = tnet.b{2};
cnet.LW{2,1} = tnet.LW{2,1};
```

Para realizar la implementación con el módulo de velocidad, se enlaza el controlador PI FUZZY al modelo experimental, del cual se obtienen las señales del sistema controlado (figura 33).



**Figura 33: Señal de salida para el sistema controlado por un controlador neuronal.**

## IX. CONCLUSIONES

La unidad de velocidad DC Speed Control System es una herramienta que permitió realizar la implementación de las diferentes técnicas de control generando respuestas aceptables.

Los algoritmos de optimización como el fminsearch, son técnicas poco usadas al momento de realizar un controlador, no obstante al implementarlo en los controladores PI, PID y PIFuzzy han generando respuestas muy aceptables.

El control predictivo, neuronal y Pifuzzy son técnicas de control inteligente que gracias a las herramientas de Matlab no fue necesario realizar un análisis matemático detallado de cada una de las técnicas.

Para el controlador por redes neuronales, se observó que se debe tener un modelo matemático que genere una dinámica aproximada a la que genera la unidad de velocidad.

Al comparar las salidas de todos los sistemas, es notorio que el sistema es más estable cuando se lo controla con los controladores digital y neuronal, ya que ambos controladores utilizan un sistema de primer orden para su diseño.

## REFERENCIAS

[1] Hossein Arsham. “Modelos Deterministas: Optimización Lineal”, 1994

[2] Optimization Toolbox For Use with MATLAB®. “Optimization Toolbox User’s Guide”. Version 3.

[3]  
[http://www.ramonmedina.name/files/todec\\_g.pdf](http://www.ramonmedina.name/files/todec_g.pdf).

[4] Enrique Castillo, Antonio J. Conejo, Pablo Pedregal, Ricardo García y Natalia Alguacil “Formulación y Resolución de Modelos de Programación Matemática en Ingeniería y Ciencia”. 20 de febrero de 2002.

[5] Consideraciones para el análisis y diseño de Sistemas de control

## BIOGRAFÍA DE AUTORES



César Andrés Villacís Fiallos nace en Ambato el 25 de Enero de 1987, sus estudios secundarios los realizó en el I.S.T.D. “GUAYAQUIL” obteniendo el título de bachiller técnico en la especialidad de Electrónica.

Actualmente ha terminado sus estudios superiores en Ingeniería Electrónica en la ESPE, sus intereses son el control inteligente, la biomédica y la robótica.



**R. Javier González P.** nació en Quito, Ecuador, un 14 de Diciembre de 1985 inicio su vida estudiantil cursando sus estudios secundarios en el Colegio Militar “Eloy Alfaro”, posteriormente realiza sus estudios de pregrado en la Escuela Politécnica del Ejército en la carrera de Automatización y Control. Entre los campos de interés esta la línea de Redes realizando cursos en la línea CISCO, Idiomas: Portugués, Inglés.