

# CONTROL PI ADAPTIVO POR GANANCIAS TABULADAS CON PLANIFICADOR DIFUSO TIPO SUGENO APLICADO EN UNA PLANTA PAR MOTOR-GENERADOR

Carlos Andrés Monsalve Rubio<sup>1</sup>, Juan Esteban Agudelo Agudelo<sup>2</sup>, Juan Camilo Gómez Cadavid<sup>3</sup>

<sup>1</sup> Estudiante Ingeniería en Instrumentación y control. andres.0490@hotmail.com

<sup>2</sup> Estudiante Ingeniería en Instrumentación y control. juan\_agudelo91103@elpoli.edu.co

<sup>3</sup> Ing. Instrumentación y Control, M.Sc. (c). Universidad de Antioquia. kmilo17pet@gmail.com

<sup>1,2 y 3</sup> Politécnico Colombiano Jaime Isaza Cadavid, carrera 48 No 7-151, Medellín, Colombia.

## RESUMEN

Se presenta un método de control adaptativo usando la técnica de ganancias tabuladas. El enfoque propuesto utiliza un planificador difuso tipo Sugeno para estimar en línea parámetros de un controlador PI digital. Los experimentos son realizados en una planta par motor-generator, donde se controla el voltaje de generación, mediante la regulación del voltaje inducido en el motor. Se evalúa el desempeño del sistema de control propuesto respecto a un controlador PI convencional, usando métricas de la integral del error. Los resultados muestran un desempeño notable en torno a la estabilidad, rechazo a perturbaciones y respuesta transitoria.

**Palabras clave:** difuso, ganancias tabuladas, Sugeno, sistema de control, identificación, motor-generator, planificador.

Recibido: 23 de Octubre de 2012.

Received: October 23<sup>rd</sup>, 2012.

Aceptado: 2 de diciembre de 2012.

Accepted: December 2<sup>nd</sup>, 2012.

## ADAPTIVE PI CONTROL THROUGH THE GAIN SCHEDULE APPROACH, USINA SUGENO FUZZY PLANNIFIER FOR A MOTOR-GENERATOR PLANT

## ABSTRACT

*This paper presents an adaptive control scheme using the gain schedule method. The proposed approach uses a Sugeno-type planner to estimate online, the parameters of a digital PI controller. The experiment is realized on a motor-generator pair plant, in which the generation voltage is controlled with the regulation of the induced voltage in the motor. The performance of the proposed control system is evaluated against a conventional PI controller using error integral metrics. The results show a better performance over stability, disturbance rejection and transient response.*

**Keywords:** Gain Schedule, Sugeno, Control Algorithm, System Identification.

## 1. INTRODUCCIÓN

En un sistema no-lineal, las salidas son funciones no lineales de las entradas [1]. En la práctica, todos los sistemas físicos son no lineales, sin embargo, es posible aplicar la teoría de los sistemas Lineales Invariantes en el Tiempo (LTI), realizando la linealización del proceso alrededor de un punto de equilibrio específico [1].

La teoría de control por ganancias tabuladas (GSC ó Gain Schedule Control) fue desarrollada en la década de los 60s. Desde el principio, este método tuvo gran acogida en el control de sistemas dinámicos, específicamente en sistemas altamente no-lineales variantes o invariantes en el tiempo [2].

El principio básico del (GSC), es conmutar los parámetros de un controlador alrededor de puntos de equilibrio del sistema. No obstante, en puntos críticos, cuando se efectúa la conmutación de parámetros, el desempeño baja significativamente. [2].

En torno a esta problemática, se han desarrollado numerosos trabajos orientados a la implementación de algoritmos que realicen una interpolación por segmentos entre los puntos de equilibrio, permitiendo así, que en cada punto de operación intermedio, las ganancias del controlador varíen gradualmente [2].

Yao y Linen [3], desarrollan un controlador PID difuso mediante el método de GSC, donde proponen dos controladores difusos (un FPI y FPD) y a ambos son ajustados mediante un parámetro de ponderación con un GSC difuso.

Kratmüller en 2009 [4] utiliza un sistema tipo Sugeno para el control adaptativo de sistemas dinámicos no lineales con incertidumbre. El controlador tiene además como objetivo nulificar los efectos de las incertidumbres y obtener un desempeño asignado a priori.

En 2010 Rezek et al. [5], proponen un control difuso por GSC para operar un motor paso a paso. La estrategia propuesta, permite adaptar las ganancias del controlador para alcanzar una velocidad de respuesta deseada. Los resultados de este trabajo demuestran un desempeño superior respecto a un sistema de control convencional. En la misma fecha, Hassan et al. [6] proponen un sistema difuso similar, basándose en controladores

PI programables, para mejorar la estabilidad en un sistema de potencia.

En este artículo, se plantea un controlador por ganancias tabuladas. La técnica propuesta, utiliza un planificador difuso con motor de inferencia tipo Sugeno. El sistema descrito es implementado en el software MATLAB®. Los experimentos son efectuados en una planta de Motores (Par motor-generator) y los resultados obtenidos, son comparados con el desempeño de un controlador PI convencional.

El documento está dividido en cinco secciones: En la sección 2 se describe la planta de experimentación, en la sección 3 la identificación del sistema en diferentes puntos de operación. En la sección 4 se presentan las bases teóricas y el desarrollo del sistema de control. La sección 5 exhibe los resultados obtenidos y la sección 6 expone las conclusiones del documento.

## 2. PLANTA DE EXPERIMENTACIÓN – PLANTA PAR MOTOR-GENERADOR

El hardware (ver figura 1) está conformado por un Subsistema mecánico y otro electrónico. El primero está conformado por dos motores que se encuentran acoplados a través de una banda elástica.

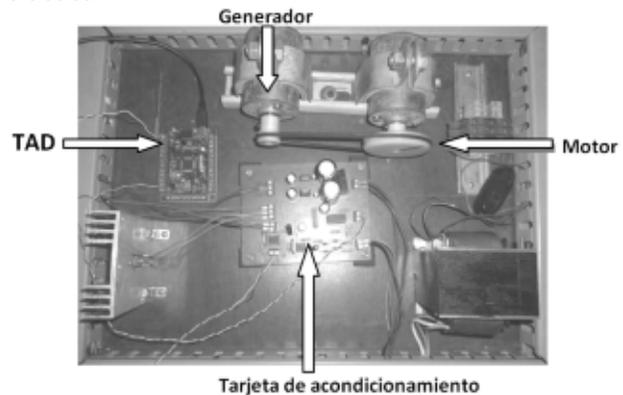


Fig.1. Planta Motor-Generador

Los motores operan a 12 voltios DC, uno de ellos funciona como motor, que aumenta o disminuye las revoluciones con un voltaje proporcional en su entrada. El segundo motor, es utilizado como un generador de voltaje, este posee el comportamiento inverso al motor que al aplicarle un trabajo mecánico, produce un voltaje en sus terminales.

El subsistema electrónico de la planta está conformado por una tarjeta de acondicionamiento de señal (ver figura 3) y un dispositivo de adquisición de datos (DAQ – PTAD-S01 Basic) (ver figura 2).

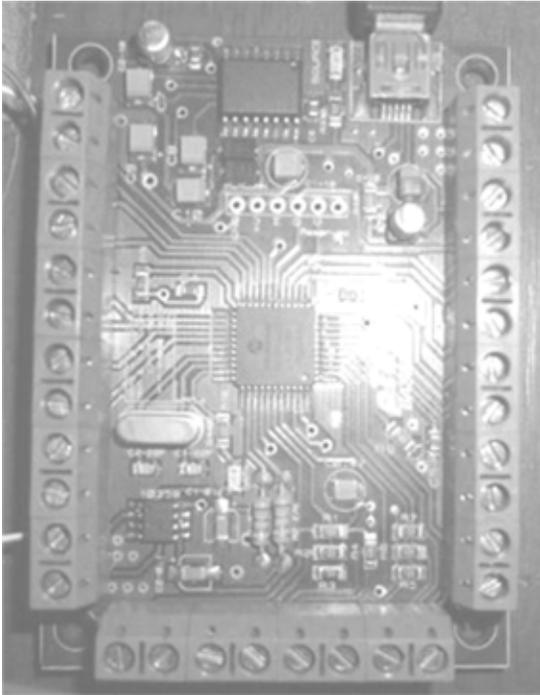


Fig.2. Tarjeta de Adquisición de Datos (TAD)

Por medio del dispositivo DAQ se establece una comunicación entre el proceso y un computador, donde se encuentra programado el sistema de control.

En la figura 4 se muestra el circuito de potencia. Este circuito recibe la señal de salida de la TAD y lo

convierte en un voltaje análogo que es el aplicado al motor.

En la figura 5 se muestra un diagrama esquemático de la planta par-motor generador. En el que se identifican claramente cada elemento del sistema y como están relacionados: computador, circuito de potencia, el motor y el generador.

### 3. IDENTIFICACIÓN DEL SISTEMA

El sistema fue sometido a una serie de entradas tipo escalón de carácter ascendente, con el fin de generar la base de datos mostrada en la figura 6. La base de datos conserva un registro del tiempo  $t$ , estímulo realizado  $u(t)$  y respuesta del sistema  $y(t)$ .

Para los datos mostrados, se implementa un algoritmo de identificación que examina los valores correspondientes a la entrada  $u(t)$  y su respectiva respuesta  $y(t)$ , es decir, se analiza cada escalón aplicado y la respuesta correspondiente de la planta. Luego, separa los datos del punto de operación encontrado y estima un modelo de primer orden con retardo descrito por la ecuación (1).

$$G(s) = \frac{ke^{-\theta s}}{\tau s + 1} \quad (1)$$

Donde  $k$  es la ganancia del sistema,  $\tau$  la constante de tiempo y  $\theta$  el tiempo muerto o retardo de la planta.

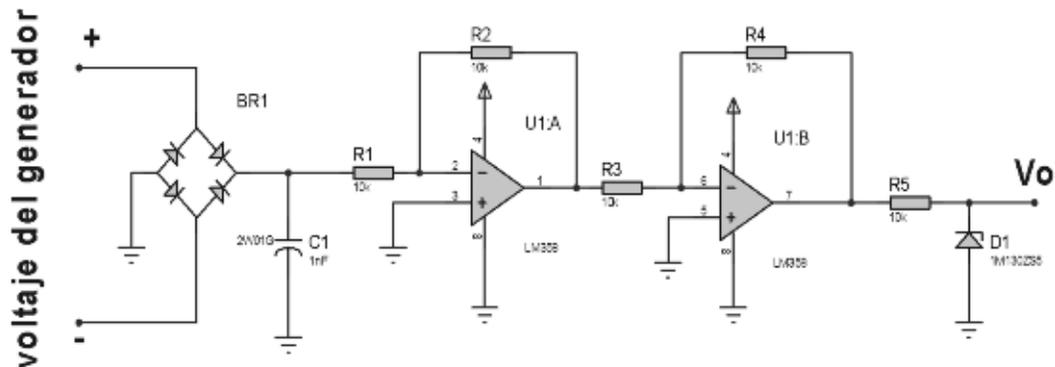


Fig. 3. Circuito de Acondicionamiento de Señal.

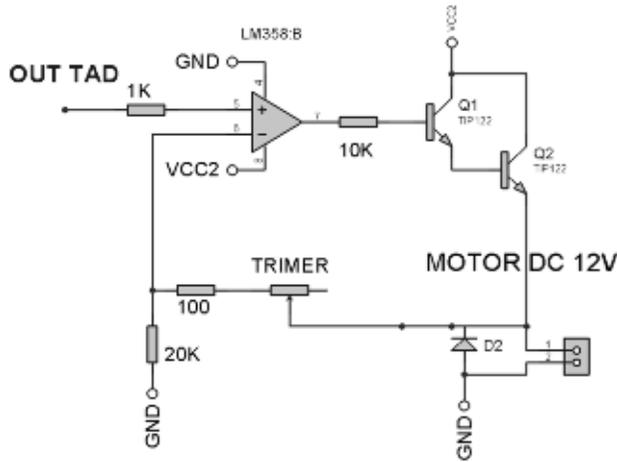


Fig. 4. Circuito de Potencia

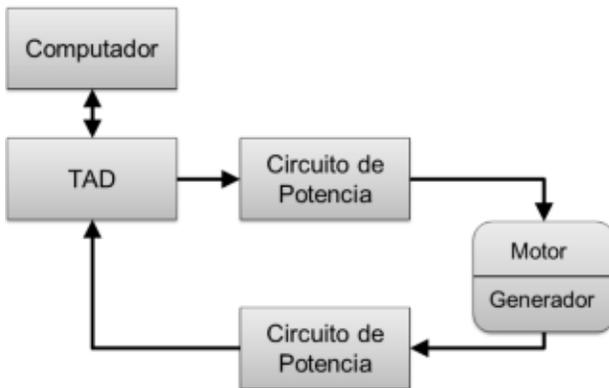


Fig. 5. Diagrama Esquemático del proceso.

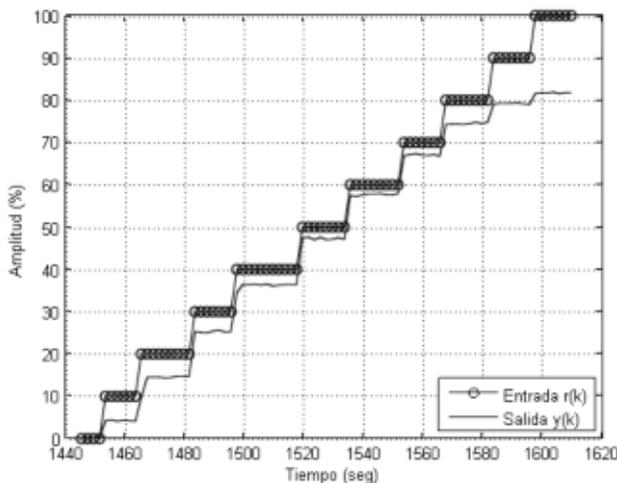


Fig. 6. Base de datos obtenida de la planta

Para los datos mostrados, se implementa un algoritmo de identificación que examina los valores correspondientes a la entrada  $u(t)$  y su respectiva

respuesta  $y(t)$ , es decir, se analiza cada escalón aplicado y la respuesta correspondiente de la planta. Luego, separa los datos del punto de operación encontrado y estima un modelo de primer orden con retardo descrito por la ecuación (1).

$$G(s) = \frac{ke^{-\theta s}}{\tau s + 1} \quad (1)$$

Donde  $k$  es la ganancia del sistema,  $\tau$  la constante de tiempo y  $\theta$  el tiempo muerto o retardo de la planta.

El algoritmo de identificación se muestra en el siguiente pseudocódigo:

```

1  Cargar base de datos.
2  Identificar valores únicos de entrada  $\vec{u}_n(t)$ 
3  Parak = 1 hasta n
4  • Identificar el intervalo de tiempo  $(t_1:t_2)$  donde
    $u(t) = u_n(t)_{[k]}$ 
5  • Identificar modelo  $G_{p_i}(s)$  para la base de datos
    $(t, r(t), y(t))$  en el intervalo  $(t_1:t_2)$ 
6  • Almacenar modelo  $M_{[k]} = G_{p_i}(s)$ 
7  Fin
Donde n es el número de escalones o valores únicos
de la entrada  $u(t)$ 

```

Los modelos obtenidos, se muestran en la tabla 1, donde se aprecia el modelo correspondiente para cada escalón aplicado.

Finalmente se obtiene la media de todos los modelos obtenidos en la tabla 1, para fines de comparación, en secciones posteriores de este documento.

$$\bar{G}_p(s) = \frac{0.8140e^{-0.061s}}{0.3288s + 1} \quad (2)$$

#### 4. ESTRATEGIA DE CONTROL

##### 4.1 Controlador por ganancia programable (GSC)

Esta técnica, es un acercamiento al control de sistemas no lineales y asume que el sistema puede representarse mediante un modelo parametrizado por ciertas variables llamadas "Scheduling Variables" (Variables de programación). Así, cuando estas variables asumen un valor constante, se obtiene un punto de operación [12].

Tabla 1: Modelos Identificados

$u(t)$	Modelo
10%	$G_{p_1}(s) = \frac{0.4268e^{-0.06s}}{0.35s + 1}$
20%	$G_{p_2}(s) = \frac{1.023e^{-0.07s}}{0.3215s + 1}$
30%	$G_{p_3}(s) = \frac{1.074e^{-0.05s}}{0.3398s + 1}$
40%	$G_{p_4}(s) = \frac{1.097e^{-0.08s}}{0.321s + 1}$
50%	$G_{p_5}(s) = \frac{1.098e^{-0.05s}}{0.3136s + 1}$
60%	$G_{p_6}(s) = \frac{1.051e^{-0.06s}}{0.3342s + 1}$
70%	$G_{p_7}(s) = \frac{0.8934e^{-0.12s}}{0.2974s + 1}$
80%	$G_{p_8}(s) = \frac{0.748e^{-0.03s}}{0.3461s + 1}$
90%	$G_{p_9}(s) = \frac{0.4638e^{-0.07s}}{0.3252s + 1}$
100%	$G_{p_{10}}(s) = \frac{0.2651e^{-0.02s}}{0.3395s + 1}$

En el enfoque clásico, se linealiza el sistema alrededor de distintos puntos de operación de interés, obteniéndose para ellos, un grupo de modelos lineales para el cual se estima un conjunto de controladores. Luego, se implementa el esquema de control, donde se ajustan o conmutan los parámetros del controlador, acorde a la variación de las variables de programación.

Dicha tarea, donde se monitorizan las variables de programación y se ajustan o conmutan parámetros, es asignada a un bloque denominado "Planificador" [13].

### 4.2 Diseño previo

A partir de los modelos obtenidos (Tabla 1), se sintonizan controladores PI digitales descritos por la ecuación (3) usando un algoritmo de búsqueda Simplex [10].

$$D_{(z^{-1})} = \frac{U(z^{-1})}{E(z^{-1})} = \frac{q_0 + q_1z^{-1}}{1 - z^{-1}} \quad (3)$$

Donde:

$$q_0 = k_c \left[ 1 + \frac{T}{2t_i} \right] \quad (4)$$

$$q_1 = -k_c \left[ 1 - \frac{T}{2t_i} \right] \quad (5)$$

En las ecuaciones (4) y (5),  $k_p$  es la ganancia proporcional,  $k_i$  la ganancia integral y  $T$  el periodo de muestreo.

El algoritmo de búsqueda Simplex minimiza una función objetivo, estimando una solución óptima local a un problema de  $n$  variables, desplazando un polítopo de  $n + 1$  vértices en  $n$  dimensiones [11]. Para este caso particular, se utiliza la función objetivada por la ecuación (6).

$$J(e, \Delta u) = \int_0^t t |e(t)| dt + \delta \int_0^t |\Delta u(t)| dt \quad (6)$$

De (6), el primer término representa el criterio del *área temporizada del valor absoluto del error (IAET)* y el segundo, el criterio del *área producida por el valor absoluto del esfuerzo de control (IADU)* multiplicado por un factor de ponderación  $\delta$ .

### 4.3 Algoritmo de búsqueda Simplex

El algoritmo simplex implementado, esta resumido en el siguiente algoritmo:

1	Con $\alpha, \beta$ y $\lambda$ , obtener un simplex inicial con vértices aleatorios $x_0, x_1, \dots, x_n$ y evaluar la función objetivo.
2	Reacomodar los vértices $x_0, x_1, \dots, x_n$ del simplex actual, de modo que $f_0, f_1, \dots, f_n$ queden en ordenado ascendente.
3	Calcular el punto de reflexión $x_r, f_r$
4	Si $f_r < f_0$
4.1	Calcular punto de expansión $x_e, f_e$
4.2	Si $f_e < f_0$ reemplazar el peor punto por el punto de expansión : $x_n = x_e, f_n = f_e$
4.3	Si $f_e > f_0$ reemplazar el peor punto por el punto de reflexión : $x_n = x_r, f_n = f_r$
5	Si $f_r > f_0$
5.1	Si $f_r < f_0$ reemplazar el peor punto por el punto de reflexión : $x_n = x_r, f_n = f_r$
5.2	Si $f_r > f_0$
5.2.1	Si $f_r > f_n$ calcular el punto de contracción $x_c, f_c$
	<ul style="list-style-type: none"> <li>• Si <math>f_c &gt; f_n</math> contraer el simplex</li> <li>• Si <math>f_c &lt; f_n</math> reemplazar el peor punto por el punto de contracción: <math>x_n = x_c, f_n = f_c</math></li> </ul>
5.2.2	Si $f_r < f_n$ reemplazar el peor punto por el punto de reflexión : $x_n = x_r, f_n = f_r$
6	Si las condiciones de parada no son satisfactorias, retornar al paso 2.

Del algoritmo anterior,  $\alpha$  es el coeficiente de reflexión y define a que distancia estará el punto de reflexión respecto al centroide del simplex;  $\beta$  es el

coeficiente de contracción, y determina a que distancia estarán los puntos de contracción cuando son contraídos del peor punto y el punto de reflexión, en el caso donde los valores de la función en dichos puntos sean iguales;  $\lambda$  corresponde al coeficiente de expansión, y define a que distancia expandir el punto reflejado en el caso donde el simplex se mueve en la dirección correcta. Los tres parámetros, también determinan la velocidad de convergencia del algoritmo [9].

La principal característica de este algoritmo, radica en que es un método de búsqueda directa sin restricciones, usando solo valores de la función objetivo, sin requerir ninguna información de su gradiente [10]. Debido a que este algoritmo presenta altos descensos durante sus primeras iteraciones, es válido para su aplicación en sistemas de control, donde simplemente se pretende encontrar parámetros que mejoren el rendimiento del lazo de control [14].

Generalizando, esta técnica comienza con un simplex, especificado por sus  $n + 1$  puntos y los respectivos valores de la función asociada. Después de obtener puntos de prueba para evaluar la función en dichos puntos, se genera un nuevo simplex hasta el punto donde valores de la función en los vértices, satisfagan la condición de descenso con respecto al simplex anterior.

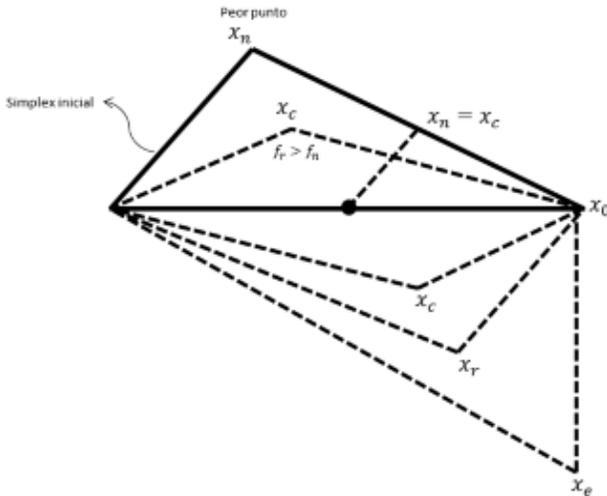


Fig. 7. Puntos que calcula el algoritmo en base al simplex inicial.

En la tabla 2 se muestran los parámetros  $q_0$  y  $q_1$  resultantes para cada modelo.

Tabla 2: Parámetros del controlador optimizados por escalón.

$u(t)$	Parámetros del PI	
	$q_0$	$q_1$
10%	1,5085	-1,1364
20%	0,59785	-0,43977
30%	0,61436	-0,45881
40%	0,5444	-0,4004
50%	0,58389	-0,42611
60%	0,60393	-0,44882
70%	0,75805	-0,54267
80%	0,92475	-0,69338
90%	1,3174	-0,9728
100%	2,6483	-1,981

#### 4.4 Planificador difuso

El planificador utiliza la inferencia difusa tipo Sugeno para asignar parámetros al controlador PI, de modo que interpole progresivamente un controlador específico dependiendo del punto de operación [2]. Para el diseño, se consideraron funciones de pertenencia (MF) gaussianas y sigmoides, con un solapamiento máximo de 0.8 en sus vecindades.

$$hgt(\mu_i \cap \mu_{i\pm 1}) \leq 4/5 \quad (7)$$

El centro de cada conjunto difuso o valor modal de las MF, está dado como  $m_i^j$  y representa los puntos de operación seleccionados para el sistema. Se tomaron como variables de programación la salida del sistema en régimen permanente y la referencia del lazo de control.

Usando la operación AND como producto, consecuente de orden cero (constante) y un congresor de tipo "combinador-lineal promedio", el proceso de inferencia para dos entradas puede ser representado por la ecuación (8) [7]:

$$f(x) = \frac{\sum_{j_1=1}^{N_1} \sum_{j_2=1}^{N_2} y^{-j_1 j_2} (\mu_{j_1}^1(x_1) \mu_{j_2}^2(x_2))}{\sum_{j_1=1}^{N_1} \sum_{j_2=1}^{N_2} (\mu_{j_1}^1(x_1) \mu_{j_2}^2(x_2))} \quad (8)$$

Donde  $x \in \mathbb{R}^2$  y  $x_1, x_2$  sus componentes,  $\mu_j^i$  es la MF para el conjunto  $A_j^i$  definido en el universo de la  $i$ -entrada en el vector  $x$ ,  $N_{1,2}$  los conjuntos difusos de la entrada y de la salida respectivamente y

$y^{-j_1j_2}$  representa el valor concreto del consecuente de la regla:

**If**  $x_1$  **is**  $A_{j_1}^1$  **AND**  $x_2$  **is**  $A_{j_2}^1$  **THEN**  $f(x)$  **is**  $y^{-j_1j_2}$

Una representación más general es dada por:

$$f(x) = \frac{\sum_{l=1}^L y^{-l} (\prod_{i=1}^N \mu_i^l(x_i))}{\sum_{i=1}^N (\prod_{i=1}^N \mu_i^l(x_i))} \quad (9)$$

Donde  $x \in \mathfrak{R}^N$ ,  $L$  es el número de reglas y  $x_i$  representa cada componente de  $x$ .

En total, se plantearon 10 funciones de pertenencia para la entrada de referencia y salida del sistema, tal como se muestra en las figuras 8 y 9 respectivamente.

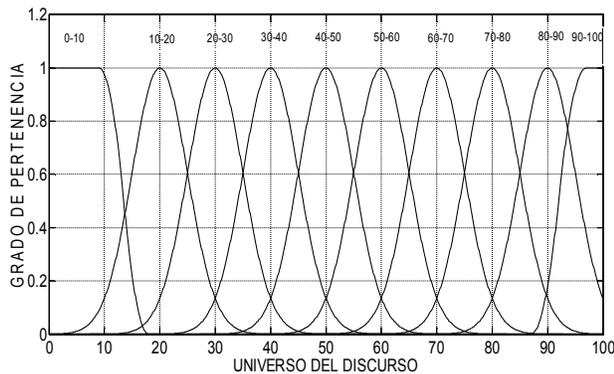


Fig. 8. Funciones de pertenencia de la referencia

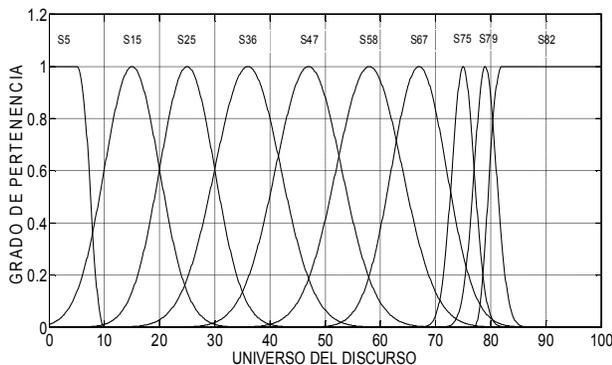


Fig. 9. Funciones de pertenencia de la salida del sistema

Las reglas planteadas son:

1. **If**  $r(t)$  **is** (0%-10%) **AND**  $y(t)$  **is** (S5%) **THEN**  $q_0 = 1.508$  **AND**  $q_1 = -1.1364$

2. **If**  $r(t)$  **is** (10%-20%) **AND**  $y(t)$  **is** (S15%) **THEN**  $q_0 = 0.5978$  **AND**  $q_1 = -0.4397$

3. **If**  $r(t)$  **is** (20%-30%) **AND**  $y(t)$  **is** (S25%) **THEN**  $q_0 = 0.6143$  **AND**  $q_1 = -0.4588$

4. **If**  $r(t)$  **is** (30%-40%) **AND**  $y(t)$  **is** (S36%) **THEN**  $q_0 = 0.5444$  **AND**  $q_1 = -0.4004$

5. **If**  $r(t)$  **is** (40%-50%) **AND**  $y(t)$  **is** (S47%) **THEN**  $q_0 = 0.839$  **AND**  $q_1 = -0.4261$

6. **If**  $r(t)$  **is** (50%-60%) **AND**  $y(t)$  **is** (S58%) **THEN**  $q_0 = 0.6039$  **AND**  $q_1 = -0.4488$

7. **If**  $r(t)$  **is** (60%-70%) **AND**  $y(t)$  **is** (S67%) **THEN**  $q_0 = 0.75805$  **AND**  $q_1 = -0.5427$

8. **If**  $r(t)$  **is** (70%-80%) **AND**  $y(t)$  **is** (S75%) **THEN**  $q_0 = 0.9247$  **AND**  $q_1 = -0.6934$

9. **If**  $r(t)$  **is** (80%-90%) **AND**  $y(t)$  **is** (S79%) **THEN**  $q_0 = 1.7431$  **AND**  $q_1 = -0.9728$

10. **If**  $r(t)$  **is** (90%-100%) **AND**  $y(t)$  **is** (S82%) **THEN**  $q_0 = 1.7724$  **AND**  $q_1 = -1.3085$

La figura 10 ilustra el esquema de control GSC planteado con el planificador difuso, encargado de variar los parámetros del controlador PI, en función de las variables de programación.

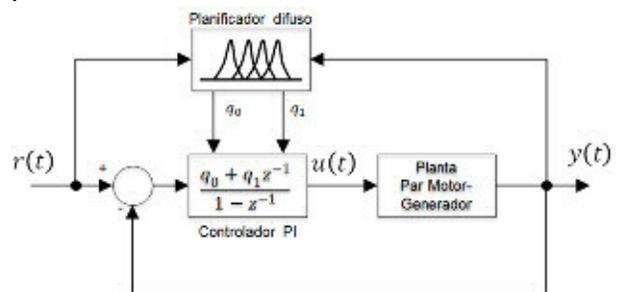


Fig. 10. Sistema de control GSC con planificador difuso

#### 4.5 Control convencional PI

En controladores PI, la señal de salida experimenta un cambio proporcional al error que actúa sobre la planta y genera una variación proporcional al cambio del dicho error [8], en (10) se muestra la ecuación diferencial de este tipo de controlador.

$$m(t) = k_p * e(t) + k_i * \int e(t) dt \quad (10)$$

Donde  $m(t)$  la señal de control y  $e(t)$  el error en el tiempo.

Aplicando transformada de *LaPlace* a (10), con condiciones iniciales iguales a cero, se obtiene la función de transferencia del controlador (11).

$$G_c(s) = \frac{M(s)}{E(s)} = \frac{k_p s + k_i}{s} \quad (11)$$

Al modelo medio de la planta  $\overline{G}_p(s)$  (Ver ecuación (2)), se le sintoniza un controlador usando la técnica propuesta por Aström y Hågglund [15] y sus ganancias fueron optimizadas usando el mismo algoritmo de búsqueda Simplex, obteniendo el controlador:

$$G_c(z) = \frac{1.645 - 1.298z^{-1}}{1 - z^{-1}} \quad (7)$$

## 5. RESULTADOS

En la figura 11 se muestra la respuesta del planificador implementado en la planta real, ante una serie de referencias  $r(t)$  en todo el rango de operación del proceso.

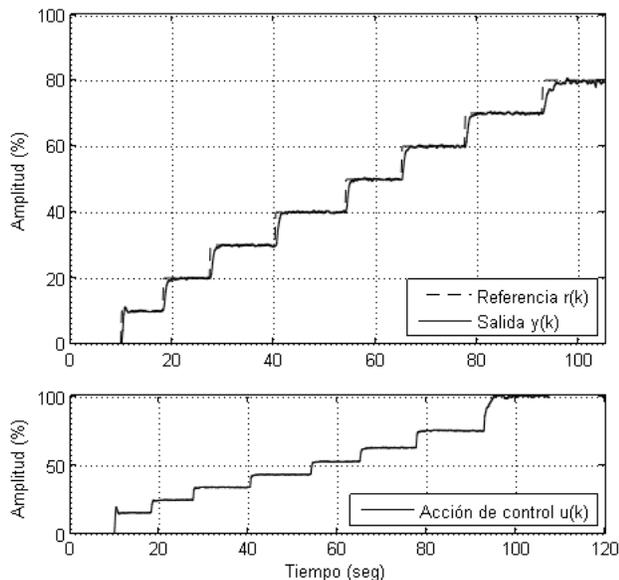


Fig. 11. Salida de la planta ante el planificador por ganancias tabuladas con programador difuso tipo Sugeno.

De manera adicional se probó el controlador añadiéndole perturbaciones a la planta. Las

perturbaciones introducidas añadiendo carga eléctrica en el motor-generador.

En la figura 12, se puede observar que las perturbaciones fueron introducidas en los instantes de tiempo (338, 345, 356, 358 y 362) segundos, donde el controlador siempre mantuvo la referencia en régimen permanente.

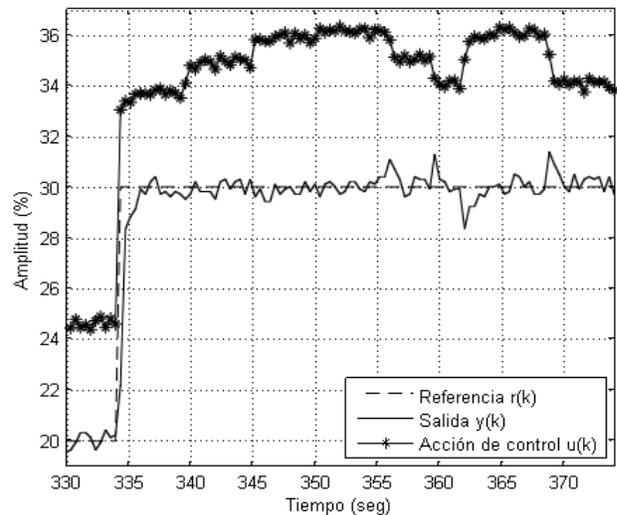


Fig. 12. salida del controlador por ganancia programable con perturbaciones.

De igual manera, el controlador clásico PI, fue sometido a una serie de escalones (ver figura 13) para apreciar su comportamiento.

Puede observarse que el controlador convencional, también elimina el error en las referencias menores al 60%, pero la salida del sistema presenta altos sobre-impulsos. En el escalón del 70%, el controlador convencional no es capaz de mantener la planta, sobre un margen aceptable, ya que en dicho punto la planta comienza a presentar la mayor no-linealidad. En este instante los parámetros fijos del controlador no son lo más adecuados para el proceso y se requiere de un nuevo controlador.

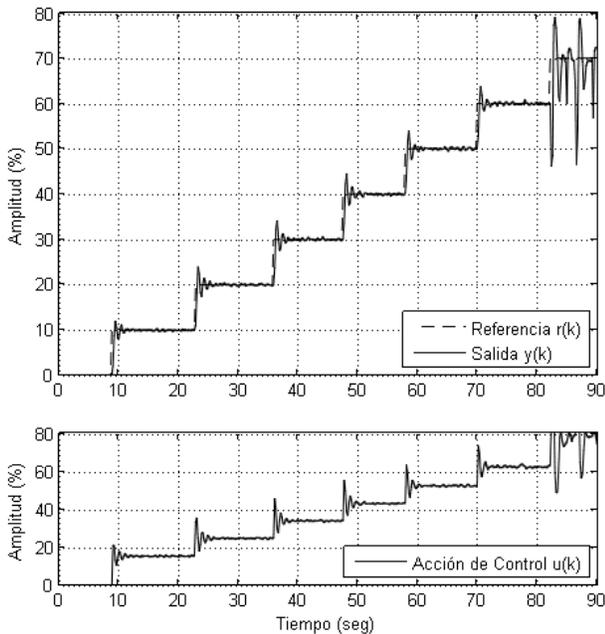


Fig. 13. salida del sistema con un controlador PI convencional

La figura 14 muestra la respuesta real del sistema con un controlador PI discreto, aplicando un escalón del 10 % y sometido a perturbaciones.

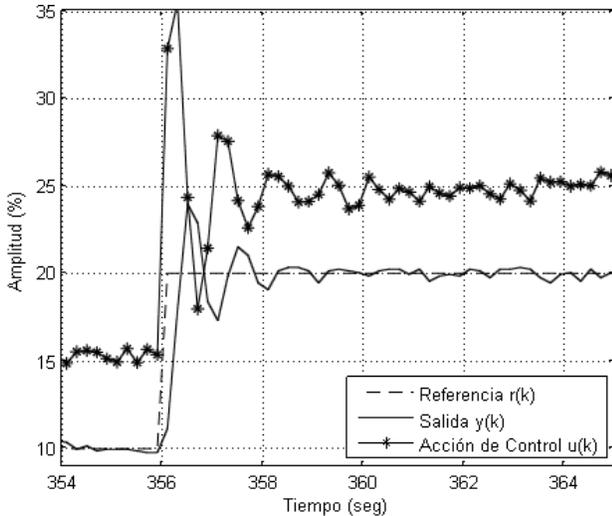


Fig. 14. Salida del sistema con controlador PI convencional sometido a perturbaciones

Tabla 3: Desempeño de los dos Controladores, Tiempo de establecimiento y Sobreimpulso Máximo

$r(\%)$	$t_s$		$M_p$	
	Ganancias tabuladas difusas	Controlador PI Convencional	Ganancias tabuladas difusas	Controlador PI Convencional
10	1,2000	2,4000	11,2000	20,0000
20	1,6000	3,1000	0,0000	40,0000
30	2,4000	2,5000	0,0000	42,0000
40	1,7000	3,9000	0,0000	45,0000
50	1,7000	2,6000	0,0000	39,5000
60	1,8000	3,6000	0,0000	37,0000
70	1,3000	No Aplica	0,0000	No Aplica
80	4,6000	No Aplica	0,0000	No Aplica

La respuesta del sistema con controlador convencional PI sometido ante perturbaciones, mostró que para escalones inferiores al 60%, el controlador es capaz de eliminar error, seguir la referencia y rechazar las perturbaciones introducidas.

Se procedió a tomar los valores aproximados del tiempo de establecimiento  $t_s$  de los dos controladores y el sobreimpulso máximo  $M_p$  para analizar el desempeño de los dos controladores. (Tabla 3).

Puede apreciarse que el controlador difuso por ganancias tabuladas presenta un menor tiempo de establecimiento que el obtenido en el controlador PI convencional. Adicionalmente, el controlador difuso no presenta sobreimpulso en la mayoría de sus referencias.

Adicionalmente, para resultados más objetivos y concluyentes, se obtienen métricas de desempeño con base en los criterios del error *Integral del valor Absoluto del Error* (IAE) (Tabla 4), *Integral del valor Absoluto del Error Temporizada* (IAET) (Tabla 5) e *Integral del error cuadrático Temporizado* (ITSE) (Tabla 6).

Tabla 4: Métrica de desempeño IAE para los Controladores Diseñados

$r(t)$	IAE	
%	Ganancias tabuladas difusas	Controlador PI Convencional
10	205,28	205,87
20	557,67	600,00
30	689,83	600,00
40	1498,53	1540,18
50	1240,47	1280,06
60	2094,13	2137,44

Tabla 5: Métrica de desempeño IAET para los Controladores Diseñados

$r(t)$	IAET	
%	Ganancias tabuladas difusas	Controlador PI Convencional
10	2303,72	2305,34
20	10995,29	11789,16
30	19777,97	20907,08
40	63179,51	64877,98
50	68463,59	70618,76
60	139658,10	142495,88

Tabla 6: Métrica de desempeño ITSE para los Controladores Diseñados

$r(t)$	ITSE	
%	Ganancias tabuladas difusas	Controlador PI Convencional
10	22604,57	22588,35
20	204182,11	233904,10
30	554138,98	619228,58
40	2444306,83	2576013,47
50	3283526,17	3492201,14
60	8148948,95	8480524,86

Finalmente, se obtienen métricas de desempeño que miden el esfuerzo de control: *Integral del valor Absoluto del cambio en el esfuerzo de control* (IADU) (Tabla 7), *Integral del valor Absoluto del cambio en el esfuerzo de control Temporizada* (ITADU) (Tabla 8).

Tabla 7: Métrica de desempeño IADU para los Controladores Diseñados

$r(t)$	IADU	
%	Ganancias tabuladas difusas	Controlador PI Convencional
10	238,8	8205,502593
20	750,2	10436,42806
30	925	13117,03233
40	1922,7	9815,195845
50	1535,3	10199,02144
60	2575,8	11368,38135

Tabla 8: Métrica de desempeño IATDU para los Controladores Diseñados

$r(t)$	IATDU	
%	Ganancias tabuladas difusas	Controlador PI Convencional
10	2650,24	86008,20621
20	14728,72	196233,1254
30	26468	368505,0939
40	80995,07	404262,2739
50	84686,42	556868,6412
60	171715,92	748369,7004

Estas métricas son fundamentales para determinar la eficiencia del controlador respecto a su tiempo de respuesta, capacidad de corrección de error y consumo energético.

## 6. CONCLUSIONES

- Se diseñó un controlador difuso por ganancias tabuladas con planificador difuso Sugeno y controlador convencional PI. La implementación de ambos mostraron buenos resultados en estabilidad, sobreimpulso y acción de control.
- Al someter la planta ante una serie de referencias ascendentes, se observa que el controlador difuso con ganancias tabuladas presenta una mejor respuesta, en cuanto a tiempo de establecimiento y sobreimpulso máximo, respecto al controlador PI convencional.
- El controlador difuso por ganancias programables presenta un mejor índice de desempeño en los criterios del error y en el esfuerzo sobre el elemento final de control. Lo anteriormente descrito se debe a que el sistema

difuso interpola los parámetros del controlador optimizados; Mientras que el controlador convencional solo tiene los parámetros optimizados de la planta promedio.

- El controlador convencional vuelve inestable a la planta en los escalones mayores al 60%, debido a que esta técnica de control no es adaptiva y el sistema es altamente no lineal. Este sistema requiere un controlador con parámetros variables, entonces los parámetros constantes del controlador no son suficientes para controlar la planta en todos los puntos de operación.

- Los sistemas no lineales se deben de controlar con técnicas de control adaptivas, ya que estas permiten variar los parámetros del controlador para cada punto de operación. Se recomienda implementar la técnica de control adaptivo con ganancias tabuladas con programador Sugeno, porque esta técnica presentó muy buenos índices de desempeño.

## 7. AGRADECIMIENTOS

Los autores expresan sus agradecimientos al Politécnico Colombiano Jaime Isaza Cadavid, por facilitar los laboratorios e instrumentos necesarios, para realizar este trabajo.

## 8. REFERENCIAS BIBLIOGRÁFICAS

[1] Ogunfunmi, T., Adaptive Non-Linear System Identification, Springer, 2007.

[2] Leith, D.J. y Leithead, W.E., Survey of Gain-Scheduling Analysis & Design, International Journal of control, vol. 73, no. 11, 1001-1025, 2000.

[3] Yao, L. y Lin, C. C., Design of Gain Scheduled Fuzzy PID Controller, World Academy of Science, Engineering and Technology 1, 2007.

[4] Kratmüller, M., The Adaptive Control of Nonlinear Systems using the T-S-K Fuzzy Logic, Acta Polytechnica Hungarica, vol. 6, no. 2, 5-16, 2009.

[5] Rezeka, S. F., Elsodany. N. M., Maharem, N. A., Fuzzy Gain Scheduling Control of a Stepper Motor Driving a Flexible Rotor, European Journal of Scientific Research, vol. 39, no. 1, 50-63, 2010.

[6] Hassan, L. H., Moghavvemi. M., Mohamed, H. A. F., Takagi-Sugeno Fuzzy Gain Scheduled PI Controller for Enhancement of Power System Stability, American Journal of Applied Sciences, vol. 7, no. 1, 145-152, 2010.

[7] Álvarez, H., Peña, M., Modelamiento de Sistemas de Inferencia Borrosa Tipo Takagi-Sugeno, Revista Avances en Sistemas e Informática, vol.6, no.1, 1-11, 2004.

[8] García, L. E., Control Digital, Teoría y práctica, Ed. Politécnico Jaime Isaza Cadavid, Politécnico Jaime Isaza Cadavid

[9] Nam Pham, Bogdan M. Wilamowski, Improved Nelder Mead's Simplex Method and Applications, Journal of Computing, Volume 3, Issue 3, march 2011, ISSN 2151-9617, 2011.

[10] J. A. Nelder, R. Mead, "A Simplex Method for Function Minimization", *Computer Journal*, vol. 7, pp. 308-313, 1965.

[11] F. Gao, L. Han, "Implementing the Nelder-Mead Simplex Algorithm with Adaptive Parameters", *Comput. Optim. Appl.*, vol., no. , pp. , 4 th May 2010.

[12] Isermann, R. *Digital Control Systems*, Springer Verlag .1981.

[13] Alzate A. et al. "Estabilizador del sistema de potencia usando ganancia programable con redes neuronales". IEEE ANDESCON99 Venezuela 1999. Memories 1er IEEE Andean Region International Conference (IEEE-Andescon99), ISBN: 078039945, 1999.

[14] Lei Yang; Xiangyang Liu; Ningfei Wang; Feng Wang, The structure analysis and design of a new self-optimizing fuzzy controller based on Nelder-Mead simplex method, Power Engineering and Automation Conference (PEAM), 2011 IEEE , Volume 3, pp 136-139, 2011.

[15] Åström K, Panagopoulos H, Hägglund T. Design of PI controllers based on non-convex optimization. *Automatica*.1998; 34:585-601.