

DISEÑO DE CONTROLADORES PI Y PID FRACCIONARIOS PARA REGULAR LA VELOCIDAD DE UN MOTOR DC UTILIZADO PARA MANEJAR UN GENERADOR DE ENERGÍA

Luis Eduardo García Jaimes¹, Carlos Andrés Velásquez Ospina², Héctor Wilson López Espinosa³

¹Magister en Educación Universidad de Antioquia, Especialista en Automatización Industrial Universidad del Valle. E-mail: legarcia47@gmail.com

²Ing. en Instrumentación y Control. Estudiante de maestría en informática énfasis en automatización y robótica de la universidad católica del norte, Antofagasta – Chile. E-mail: canveo@hotmail.com

³Ing. en Instrumentación y Control. Medellín - Colombia. E-mail: hwloes@hotmail.com

RESUMEN

Se presenta el diseño de un controlador PID fraccionario para regular la velocidad de un motor de DC que mueve un generador de energía eléctrica y su desempeño es comparado con el de un controlador PID convencional sintonizado utilizando el método del lugar geométrico de las raíces. El trabajo se divide en tres partes así: en la primera parte se establece el comportamiento dinámico del sistema de generación de energía formado por dos motores de DC, uno funcionando como motor propiamente dicho y otro como generador, en la segunda parte, se plantean las bases teóricas del cálculo fraccionario y del diseño de controladores PID fraccionarios y, en la tercera parte, se diseñan e implementan los algoritmos de control PID fraccionario y PID convencionales para el sistema motor-generador y se compara el desempeño de los algoritmos diseñados. Los resultados obtenidos indican que el controlador PID fraccionario puede ser una buena opción para realizar el control de procesos industriales.

Palabras clave: *sistema de control, PID fraccionario, NINTEGER, identificación, motor-generador.*

Recibido: 25 de octubre de 2014.

Aceptado: 16 de diciembre de 2014.

Received: October 25th, 2014.

Accepted: December 16th, 2014.

DESIGN OF A FRACTIONAL PID CONTROLLER FOR A DC MOTOR USED FOR DRIVING A POWER GENERATOR

ABSTRACT

The design of a fractional PID controller to regulate the speed of a DC motor driving an electric generator is presented and its performance compared with that of a conventional PID controller tuned using the root locus method. The work is divided into three parts as follows: in the first part the dynamic behavior of the system power generation consists of two DC motors, one running as the engine itself and the other as a generator, in the second part, states raised the theoretical basis of fractional calculus and fractional PID controller design and in the third part, was designed and implemented fractional PID control algorithms and conventional PID controller for motor-generator and performance system is compared. The results indicate that the fractional PID controller can be a good option for control of industrial processes.

Keywords: *control system, PID fractional, NINTEGER, identification, engine-generator.*

1. INTRODUCCIÓN

Este artículo se ha estructurado desde los fundamentos y definiciones básicas del control fraccionario hasta las estrategias de implantación de controladores, incluyendo la identificación del sistema y el diseño de los controladores.

El cálculo fraccionario es la extensión del cálculo que permite considerar la integración y la derivación de cualquier orden, no necesariamente entero. En el dominio del tiempo, los operadores derivada e integral fraccionarios vienen definidos por la operación de convolución, por lo que están especialmente indicados para describir fenómenos de memoria. En el dominio de Laplace, dichos operadores se corresponden con el operador $s^\alpha, \alpha \in \mathbb{R}$.

El uso de controladores del tipo PID fraccionarios permite, frente al PID convencional, la aplicación de hasta cinco especificaciones de diseño, ya que cuentan con cinco parámetros a sintonizar. El método de sintonía propuesto se basa en especificaciones de frecuencia de cruce de ganancia, margen de fase, robustez ante cambios en la ganancia de la planta, restricción en la función de sensibilidad y rechazo al ruido de alta frecuencia.

El objetivo del trabajo es diseñar controladores PI y PID fraccionarios para controlar la velocidad de un motor de DC utilizado para manejar un generador de energía y comparar su desempeño con el de controladores PI y PID convencionales utilizando el método del lugar geométrico de las raíces (LGR). El empleo de los controladores fraccionarios en aplicaciones industriales está en etapa de desarrollo de ahí la importancia de este trabajo. En Romero (2010), se presenta la aplicación de las herramientas de cálculo fraccionario al control predictivo basado en modelos, con la idea de extender la metodología y generalizar su formulación, de tal forma que se puedan conjugar las ventajas de ambas disciplinas. Von Borries (2012) analizó la estabilidad y realizó simulaciones de un sistema adaptable basado en el Modelo de Error, usando leyes de ajuste con la derivada fraccionaria de Caputo de orden $\alpha \in (0, 1]$, estudiando primero el caso escalar y luego, el caso vectorial. Johnson (2010) implementa y evalúa comparativamente tres estrategias de control: Control PI Tradicional, Control PI Fraccionario (FOPI) y Control adaptativo por modelo de

referencia (MRAC). La implementación se realizó en un controlador industrial NI compact Field Point programado en ambiente LabView. Hernández (2013), modela el sistema de combustión de un horno cementero rotatorio de proceso húmedo, utilizando un balance de energía y una identificación de orden fraccionario y diseña un controlador de orden fraccionario en combinación con un método específico para hallar las ganancias proporcionales usando LQR Linear Quadratic Regulator.

2. MATERIALES Y MÉTODOS

2.1 Descripción del sistema motor-generator

El sistema implementado cuenta con dos motores de corriente continua de 24V. El primero se utiliza propiamente como motor y el segundo, movido por el primero, trabaja como generador, se desea controlar el voltaje generado.

El hardware para el sistema de monitoreo y control del sistema está dispuesto como se indica en la figura 1 y lo constituyen los siguientes elementos:

- Tarjeta de adquisición de datos: P-TAD S01 BASIC-USB DAQ Multifunción, 10 bits, 25kS/s con 8 entradas análogas de 0 a 5v dos salidas análogas de 0 a 5V, 7 salidas digitales y 8 entradas digitales
- Sistema de acondicionamiento de la señal: toma el voltaje de salida del generador de 12V y lo convierte en señal de 0 a 5V
- Sistema de potencia: realiza la interfaz entre la tarjeta de adquisición de datos y el motor. Recibe señal de entrada de 0 a 5V y da una salida de 0 a 24V y 3A
- Computador.
- Generador: da una salida máxima de 24 VDC
- Motor de DC: alimentación de 24 VDC

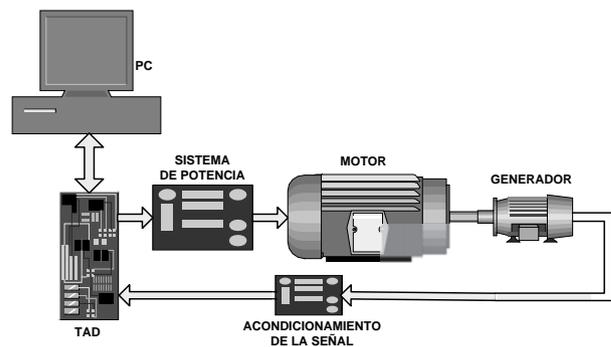


Fig. 1. Hardware para monitoreo y control del sistema

2.2 Respuestas del sistema a la señal escalón.

La figura 2 muestra la respuesta del sistema correspondiente al voltaje producido por el generador, este voltaje es proporcional a la velocidad de giro del motor y, por esta razón, es la variable que permite realizar el control de la velocidad. Para obtener la dinámica del sistema se analiza la respuesta correspondiente a cada escalón y se obtienen modelos de primer orden con retardo. Los modelos obtenidos se validan y, finalmente se promedian sus parámetros para obtener el modelo definitivo.

2.3 Identificación del sistema motor-generador.

A continuación se presentan las respuestas individuales correspondientes a cada uno de los escalones. A la respuesta de cada escalón se le hizo una traslación de ejes con el fin de referenciar a cero las señales y facilitar el proceso de identificación. Esto se puede observar en las figuras 3, 4, 5 y 6. Para la identificación se utilizó el toolbox IDENT de MATLAB.

La identificación del sistema a partir de este escalón dio como modelo:

$$G(s) = \frac{1.22e^{-0.017S}}{0.031S + 1} \quad (1)$$

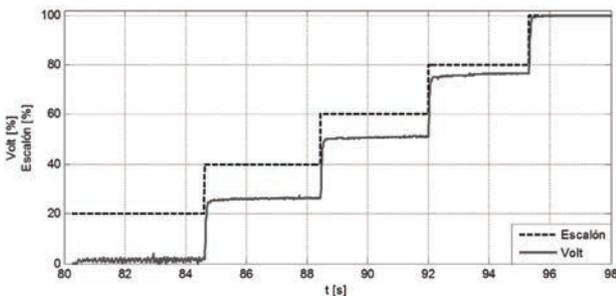


Fig. 2. Respuesta del sistema ante los estímulos

La identificación del sistema a partir de este escalón dio como modelo:

$$G(s) = \frac{1.23e^{-0.0154S}}{0.035S + 1} \quad (2)$$

La identificación del sistema a partir de este escalón dio como modelo:

$$G(s) = \frac{1.25e^{-0.016S}}{0.034S + 1} \quad (3)$$

El modelo correspondiente a este escalón es:

Planta de primer orden:

$$G(s) = \frac{1.23e^{-0.0118S}}{0.032S + 1} \quad (4)$$

Con los modelos obtenidos se estima un modelo promedio cuya discretización con un periodo de muestreo $T = 0.01$ s da como resultado:

$$G(z) = \frac{0.1732z + 0.1488}{z^3 - 0.7385z^2} \quad (5)$$

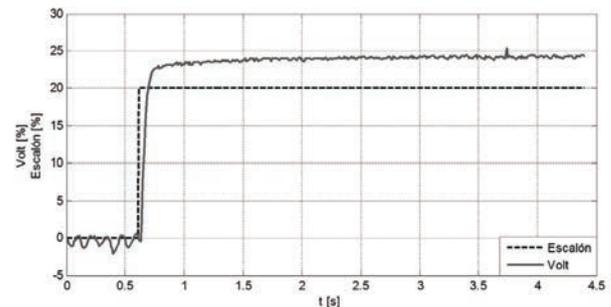


Fig. 3. Datos trasladados escalón del 20%-40%

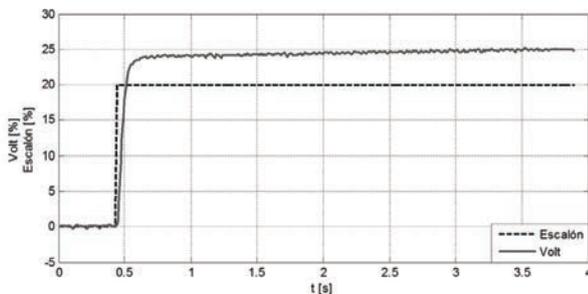


Fig. 4. Datos trasladados escalón del 40%-60% en forma de escalón

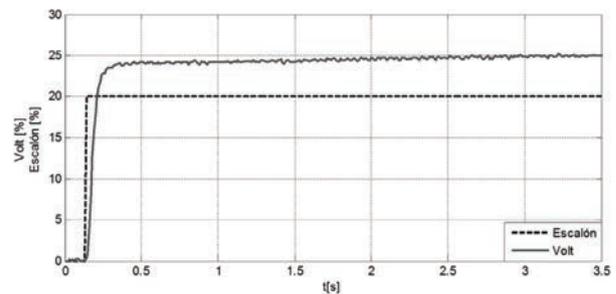


Fig. 5. Datos trasladados escalón del 60%-80% en forma de escalón

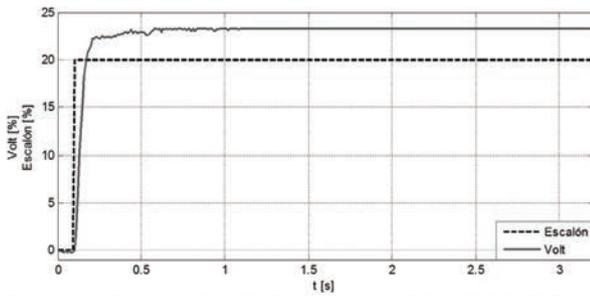


Fig. 6. Datos trasladados escalón del 80% al 100%.

2.4 Controladores PID fraccionarios

En la actualidad, los controladores industriales sólo permiten la ejecución de tres acciones básicas de control, a saber: Proporcional (P), Integral (I) y Derivativa (D) y con estas se pueden obtener controladores P, PD, PI o PID. En este trabajo se presenta la generalización de las anteriores acciones básicas de control. Esta generalización se logra utilizando operadores integro-diferenciales de orden fraccionario para obtener las acciones Derivativa e Integral de orden no entero. Al realizar esta generalización, además de los tres parámetros de ajuste de que disponen los controladores PID clásicos (la ganancia proporcional K_p , y las ganancias integral y derivativa, K_i y K_d), se añade la posibilidad de ajustar el orden de integración (λ) y de la derivada (μ), con lo que se incrementa en dos el número de parámetros susceptibles de ajuste.

El controlador PID fraccionario se formula como:

$$C(S) = \frac{M(S)}{E(S)} = K_p + \frac{K_i}{S^\lambda} + K_d S^\mu \quad (6)$$

En la figura 7 se muestra la región de trabajo de un controlador $PI^\lambda D^\mu$ fraccionario. Si $\lambda = 1$ y $\mu = 1$ se tendrá el controlador PID convencional

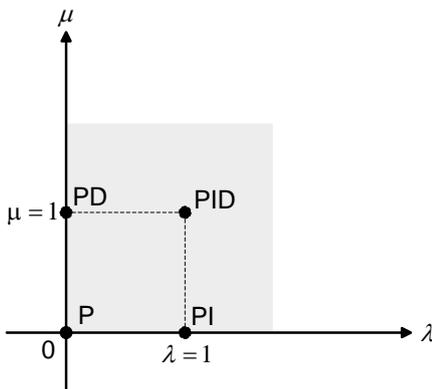


Fig. 7. Representación gráfica de un PID fraccionario (sombreado).

2.5 Sintonía de los controladores PID fraccionarios

• *Ajuste mediante el Método de respuesta en frecuencia:* este método de sintonía propuesto para estos controladores se basa en establecer especificaciones de frecuencia de cruce por cero ω_c , margen de fase ϕ_{PM} , robustez ante cambios en la ganancia de la planta, restricción en la función de sensibilidad y rechazo al ruido de alta frecuencia. Este método calcula, mediante un algoritmo iterativo, el valor de los parámetros del controlador que minimiza estas cinco funciones. Este problema de minimización no lineal se ha resuelto empleando la función *fmincon* de Matlab (Matlab 2013) y (Monje, 2004)

Los ajustes se realizan con el cumplimiento de las siguientes especificaciones:

1. La ganancia a la frecuencia de cruce de ganancia ω_c debe ser igual a 1 es decir 0 dB

$$|F(\omega_c)| = |C(\omega_c) * G_p(\omega_c)| = 1 \quad (7)$$

2. El Margen de fase ϕ_{PM} debe ser constante con valor:

$$\arg[C(\omega_c) * G_p(\omega_c)] = \phi_{PM} - \pi \quad (rad)$$

3. Rechazo al ruido en alta frecuencia ω_h en lazo cerrado debe ser

$$|T(\omega_h)| = \left| \frac{C(\omega_h) * G_p(\omega_h)}{1 + C(\omega_h) * G_p(\omega_h)} \right| < H \quad (8)$$

En donde T es la función de sensibilidad complementaria y H es la atenuación del ruido deseada para frecuencias mayores que ω_h

4. Rechazo de perturbaciones de la salida en lazo cerrado

$$|S(\omega_l)| = \left| \frac{1}{1 + C(\omega_l) * G_p(\omega_l)} \right| < N \quad (9)$$

En donde N es el valor deseado para la función de sensibilidad S para frecuencias menores que ω_l

5. Robustez ante variaciones en la ganancia de la planta

$$\frac{\partial}{\partial \omega} \arg[C(\omega_c) * G_p(\omega_c)] = 0 \quad (10)$$

En todas las expresiones anteriores $C(S)$ representa la función de transferencia del controlador y $G_p(S)$ representa la de la planta.

La función a minimizar es:

$$|F(\omega_c)| = |C(\omega_c) * G_p(\omega_c)| \quad (11)$$

- **Ajuste por tablas o Reglas de ajuste:** existen tablas con reglas de ajuste como las que se han propuesto para los controladores *PID* enteros. Éstas han sido obtenidas previamente mediante optimización fijando los valores de los parámetros para plantas del primer orden con retardo (POR) y obteniendo los cinco parámetros que definen al *PID* fraccionario. En (Valério, 2005) se pueden encontrar varias tablas resumen de estos valores de ajuste para el *PID* en función de sus parámetros. La gran utilidad de estas tablas es que pueden ser aplicadas incluso sin disponer del modelo exacto de la planta a controlar.

3 RESULTADOS

3.1 Diseño del controlador *PID* fraccionario

El modelo promedio de la planta dio como resultado:

$$G(S) = \frac{1.23e^{-0.015S}}{0.033S + 1} \quad (12)$$

La figura 8 corresponde al diagrama de bode del sistema sin el controlador y de ella se obtiene que: $\phi_{PM} = 10.5 \text{ dB}$, $MG = 126^\circ$, $\omega_c = 22 \text{ rad/s}$ y $\omega_\pi = 120 \text{ rad/s}$.

Condiciones de diseño: Para el cálculo del controlador $PI^\lambda D^\mu$, se asumen las siguientes condiciones:

- Función a minimizar:

$$|F(\omega_c)| = |C(\omega_c) * G_p(\omega_c)| \quad (13)$$
- La ganancia a la frecuencia de cruce de ganancia ω_c es igual a 1 (0 dB)

$$|F(\omega_c)| = |C(\omega_c) * G_p(\omega_c)| = 1 \quad (14)$$
- Margen de fase ϕ_{PM} debe ser de 60° (1.047 rad)

$$\text{arg}[C(\omega_c) * G_p(\omega_c)] = 1.047 - \pi \quad (rad) \quad (15)$$

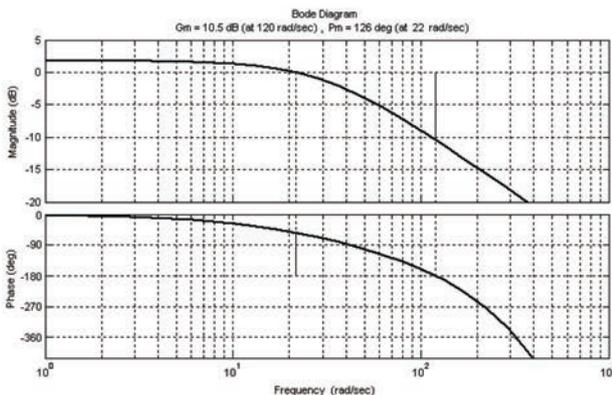


Fig. 8. Diagrama de bode del sistema sin controlador

- Rechazo al ruido en alta frecuencia ($\omega_h = 10\omega_c$) menor de 20 dB (<0.01)

$$|T(\omega_h)| = \left| \frac{C(\omega_h) * G_p(\omega_h)}{1 + C(\omega_h) * G_p(\omega_h)} \right| < 0.01 \quad (16)$$

Rechazo de perturbaciones de la salida ($\omega_l = 0.1\omega_c$) menor de 20 dB (<0.01)

$$|S(\omega_l)| = \left| \frac{1}{1 + C(\omega_l) * G_p(\omega_l)} \right| < 0.01 \quad (17)$$

En todas las expresiones anteriores $C(S)$ representa la función de transferencia del controlador y $G_p(S)$ representa la de la planta.

Haciendo $S = i\omega_c = 22 * i$ en la ecuación del controlador y en la función de transferencia del proceso y asumiendo $\mu = 0.6$ se obtiene:

$$C(i\omega_c) = K_p + \frac{K_i}{(22i)^{0.6}} + K_d(22i)^\mu \quad (18)$$

$$G(i\omega_c) = \frac{1.23e^{-0.33i}}{0.726i + 1} \quad (19)$$

De la ecuación (19) se deduce que: $|G(i\omega_c)| = 1$ y $\angle G(i\omega_c) = -36.3^\circ = 0.633 \text{ rad}$.

Haciendo $S = i\omega_h = 220 * i$ en la ecuación del controlador y en la función de transferencia del proceso y asumiendo $\mu = 0.6$ se obtiene:

$$C(i\omega_h) = K_p + \frac{K_i}{(220i)^{0.6}} + K_d(220i)^\mu \quad (20)$$

$$G(i\omega_h) = \frac{1.23e^{-3.3i}}{7.26i + 1} \quad (21)$$

De la ecuación (21) resulta: $|G(i\omega_h)| = 0.167$ y $\angle G(i\omega_h) = -54.8^\circ = -0.957 \text{ rad}$

Haciendo $S = i\omega_l = 2.2 * i$ en la ecuación del controlador y en la función de transferencia del proceso y asumiendo $\mu = 0.6$ se obtiene:

$$C(i\omega_l) = K_p + \frac{K_i}{(2.2i)^{0.6}} + K_d(2.2i)^\mu \quad (22)$$

$$G(i\omega_l) = \frac{1.23e^{-0.033i}}{0.0726i + 1} \quad (23)$$

De la ecuación (23) se obtiene: $|G(i\omega_l)| = 1.23$ y $\angle G(i\omega_l) = 6^\circ = 0.105 \text{ rad}$.

Combinando adecuadamente las ecuaciones 13 a 23 se obtienen las expresiones que conforman la función que se debe minimizar y las restricciones impuestas para dicha minimización. Al resolver las ecuaciones resultantes utilizando la *función fmincon* se obtienen los parámetros del controlador $PI^\lambda D^\mu$.

Función a minimizar:

$$|C(i\omega_c)| = \left| K_p + \frac{K_i}{(22i)^{0.6}} + K_d(22i)^\mu \right| \quad (24)$$

Restricciones:

- Ganancia a la frecuencia de cruce de ganancia ω_c :

$$h(1) = \left| K_p + \frac{K_i}{(22i)^{0.6}} + K_d(22i)^\mu \right| = 1 \quad (25)$$

- Margen de fase de 60° (1.047 rad) $\phi_{PM} = \pi + \angle G(i\omega_c) + \angle C(i\omega_c)$ de acá se deduce que $\angle C(i\omega_c) = -65.2^\circ = -1.137 \text{ rad}$

$$h(2) = \arg \left(K_p + \frac{K_i}{(22i)^{0.6}} + K_d(22i)^\mu \right) = -1.137 \quad (26)$$

- Rechazo al ruido en alta frecuencia ($\omega_h = 10\omega_c$) menor de 20 dB (<0.01)

$$g(1) = \left| \frac{0.167 * \left(K_p + \frac{K_i}{(220i)^{0.6}} + K_d(220i)^\mu \right)}{1 + 0.167 * \left(K_p + \frac{K_i}{(220i)^{0.6}} + K_d(220i)^\mu \right)} \right| < 0.0132 \quad (27)$$

- Rechazo de perturbaciones de la salida ($\omega_l = 0.1\omega_c$) menor de 20 dB (<0.01)

$$g(2) = \left| \frac{1}{1 + 1.23 * \left(K_p + \frac{K_i}{(2.2i)^{0.6}} + K_d(2.2i)^\mu \right)} \right| < 0.01 \quad (28)$$

Como puede verse, la definición del problema consta de una función a minimizar, dos igualdades y dos desigualdades, elementos suficientes para estimar los cuatro parámetros del controlador: K_p, K_i, K_d y λ necesarios para minimizar la función de costo.

El controlador estimado en Matlab con la función *fmincon* es:

$$G(S) = \frac{U(S)}{E(S)} = -0.54202 + \frac{1.1277}{S^{0.6}} + 0.24801S^{0.2} \quad (29)$$

En el toolbox de control fraccionario denominado "NINTEGER" (Valerio 2011), se proporcionan las herramientas necesarias para transformar el controlador PID fraccionario en una función de transferencia con potencias de S enteras utilizando el comando *nipid* el cual tiene la siguiente sintaxis:

`C= nipid(kp,kd,u,ki,lan,[ancho de banda],n,'fórmula')`

En donde:

kp, kd, u, ki, lan: son los parámetros del controlador.

[ancho de banda]: es el ancho de banda especificado para el controlador.

n: El número de ceros y polos de la aproximación.

'fórmula': es el tipo de aproximación utilizada. Por ejemplo 'crone' para sistemas continuos ó 'tustin' para sistemas discretos.

El código que se da a continuación permite obtener controladores continuos y/o discretos a partir de un controlador $PI^\lambda D^\mu$ fraccionario:

clc

% Aproximación entera del controlador PID fraccionario

kp=-0.54202;

ki=1.1277;

kd=0.24801;

lan=0.6;

u=0.2;

cc=nipid(kp,kd,u,ki,lan,[2 200],2,'crone')% Continuo

cd=nipid(kp,kd,u,ki,lan,0.01,'tustin') %Discreto

Como resultado se obtienen las siguientes funciones de transferencia:

Controlador PID continuo equivalente:

$$C(S) = \frac{U(S)}{E(S)} = \frac{0.07932S^3 + 1.706S^2 - 189.2S + 1782}{S^3 + 81.94S^2 + 1592S} \quad (30)$$

Controlador PID discreto equivalente con T = 0.01 s :

$$C(z) = \frac{U(z)}{E(z)} = \frac{0.2205 - 0.21017z^{-1} + 0.32095z^{-2} - 0.08445z^{-3}}{1 - z^{-1}} \quad (31)$$

3.2 Controlador PI fraccionario con ajuste por tablas.

Teniendo en cuenta los parámetros del proceso $K = 1.23$, $\tau = 0.033 \text{ s}$ y $\theta = 0.015 \text{ s}$, los ajustes del controlador son:

$$\delta = \frac{\theta}{\theta + \tau} = \frac{0.015}{0.033 + 0.015} \quad \delta = 0.3125 \quad (32)$$

$$\lambda = \begin{cases} 1.1 & \text{si } \delta \geq 0.6 \\ 1.0 & \text{si } 0.4 \leq \delta < 0.6 \\ 0.9 & \text{si } 0.1 \leq \delta < 0.4 \\ 0.7 & \text{si } \delta < 0.1 \end{cases} \quad \lambda = 0.9 \quad (33)$$

$$K_p = \frac{1}{K} \left[\frac{0.2978}{\delta + 0.00037} \right] \quad K_p = 0.7738 \quad (34)$$

$$T_i = \tau \left[\frac{0.8578}{\delta^2 - 3.4002\delta + 2.405} \right] \quad T_i = 0.01965 \quad (35)$$

$$K_i = \frac{K_p}{T_i} \quad K_i = 39.3791 \quad (36)$$

Por lo tanto, el controlador PI^λ es:

$$C(S) = 0.7738 + \frac{39.3791}{S^{0.9}} \quad (37)$$

El sistema se simuló con este controlador y se obtuvo una respuesta rápida, estable pero con sobreimpulso muy alto (>40%), por lo que se hace uso de la sugerencia propuesta en (Chen 2007), con lo cual el controlador definitivo queda así:

$$C(S) = 0.6577 + \frac{33.4722}{S^{0.9}} \quad (38)$$

Controlador PI continuo equivalente:

$$C(S) = \frac{0.6577S^2 + 69.55S + 841.8}{S^2 + 25.18S} \quad (39)$$

Controlador PI discreto equivalente con $T = 0.01$ s :

$$C(z) = \frac{U(z)}{E(z)} = \frac{0.942 - 0.4302z^{-1} - 0.0511z^{-2} + 0.0056z^{-3}}{1 - z^{-1}} \quad (40)$$

3.3 Diseño del controlador PID convencional

Para diseñar controladores PI o PID utilizando el Lugar Geométrico de la Raíces es necesario que el polo asignado cumpla dos condiciones: la condición de módulo y la condición de ángulo.

Condición de módulo:

$$|D(z)HG(z)|_{polo} = 1 \quad (41)$$

Condición de ángulo:

$$\arg(D(z)HG(z))|_{polo} = -180 \quad (42)$$

• *Diseño del controlador PI convencional.* Para un controlador PI discreto la función de transferencia se puede escribir como:

$$D(z) = \frac{q_0z + q_1}{z - 1} \quad (43)$$

La función de transferencia discreta del sistema con $T = 0.01$ es:

$$HG(z) = \frac{0.1732(z + 0.8591)}{z^2(z - 0.7385)} \quad (44)$$

La función de transferencia en lazo abierto del proceso está dada por:

$$F(z) = \frac{q_0z + q_1}{z - 1} * \frac{0.1732(z + 0.8591)}{z^2(z - 0.7385)} \quad (45)$$

El tiempo de establecimiento del sistema en lazo abierto es $t_s = 0.132$ s. Para realizar el diseño del controlador se asume que el tiempo de establecimiento del sistema en lazo cerrado es $t_{sLC} = 0.1025$ s y que el coeficiente de amortiguamiento es $\xi = 0.8$

Las condiciones de diseño establecidas ubican el polo dominante de lazo cerrado deseado en $z = 0.6488 \pm j0.1949$. Este polo debe cumplir la condición de módulo y la condición de ángulo, es decir:

$$|F(z)| = \frac{6.6607\sqrt{(q_1 + 0.6488q_0)^2 + 0.03798q_0^2}}{1} = 1 \quad (46)$$

$$\begin{aligned} \arg(F(z)) &= \arg\left(\frac{q_0z + q_1}{z - 1}\right) \\ & * \left.\frac{0.1732(z + 0.8591)}{z^2(z - 0.7385)}\right|_{z=0.6488+j0.1949} \\ &= -180 \quad (47) \end{aligned}$$

Resolviendo simultáneamente las ecuaciones (46) y (47) se obtiene: $q_0 = 0.7152$ y $q_1 = -0.5198$

Así el controlador PI es:

$$D(z) = \frac{U(z)}{E(z)} = \frac{0.7152z - 0.5198}{z - 1} \quad (48)$$

• *Diseño del controlador PID convencional.* Utilizando el método del lugar geométrico de las raíces y con polos ubicados en $z = 0.6488 \pm j0.1949$ se diseñó un controlador PID discreto de modo que se cumplieran la condición de módulo y la condición de ángulo. Como este controlador tiene tres parámetros

• por ajustar, se adicionó una tercera condición haciendo que: $q_2 = -0.25q_1$. El resultado del diseño dio el siguiente controlador PID:

$$D(z) = \frac{U(z)}{E(z)} = \frac{0.9843z^2 - 1.0851z + 0.2712}{z(z - 1)} \quad (49)$$

4 DISCUSIÓN

Implementación de los controladores PID fraccionario y PID convencional en el sistema real

Para implementar los controladores fraccionarios se hizo uso del toolbox NINTEGER de Matlab desarrollado por Valério (Valério 2011). La figura 9

muestra el diagrama en bloques utilizado para realizar el control con Simulink.

El bloque *nipid* corresponde al controlador fraccionario, el bloque *Analog output* es el bloque para la escritura de datos, el bloque *analog input* es el bloque de lectura de datos, éstos dos bloques se utilizan para realizar la comunicación entre la planta y el computador mediante la tarjeta de adquisición de datos. El bloque *PTAD INIT* habilita la tarjeta para su funcionamiento y, finalmente el bloque *PTAD-SCOPE* permite visualizar las variables que intervienen en el proceso.

Resultados obtenidos con el controlador PI y PID fraccionarios

Se calculó un controlador PID fraccionario utilizando la función *fmincon* de Matlab y un controlador PI fraccionario utilizando ajuste por tablas. El cálculo de estos controladores se presentó en el apartado anterior.

La figura 10 corresponde a la respuesta del sistema con el controlador fraccionario implementado directamente en simulink utilizando el toolbox *Ninteger*.

Se observa que el sistema se estabiliza con error promedio cero, sin sobreimpulso y la acción de la ley de control sobre el elemento final de control es suave y sin oscilaciones significativas.

En la figura 11 se presenta la respuesta del sistema con el controlador PID fraccionario en su forma equivalente de potencias en S enteras la cual se obtuvo mediante el comando *nipid* y la opción *crone*. Al igual que con el controlador PID Fraccionario original, la respuesta del sistema es estable, sin sobreimpulsos y con error promedio cero, la diferencia radica en que este es más lento presentando mayor tiempo de establecimiento.

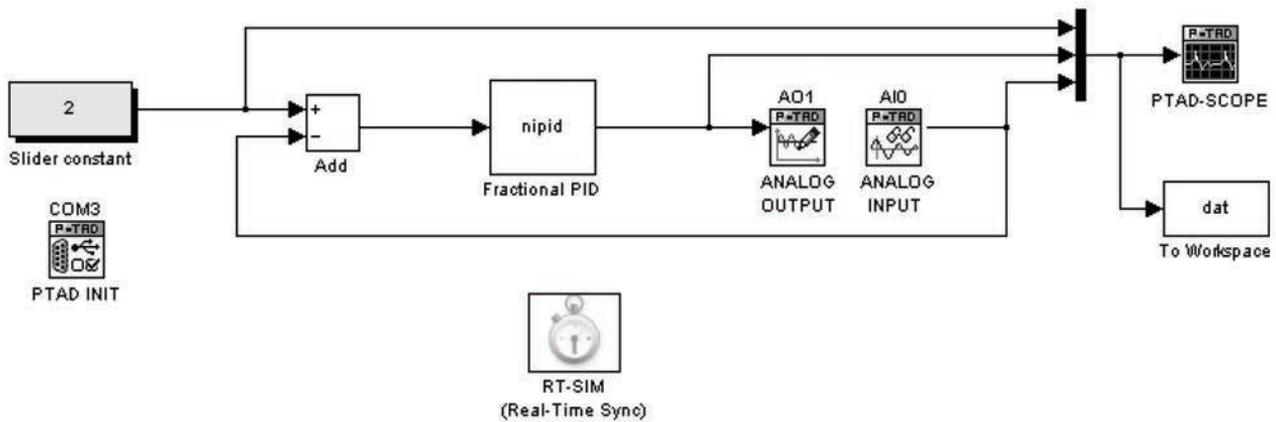


Fig. 9. Diagrama en Simulink para realizar el control fraccionario

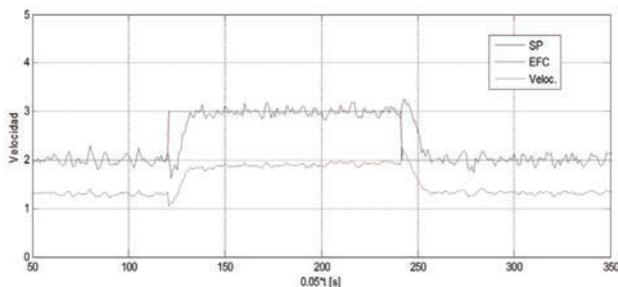


Figura 10. Respuesta con el controlador PID fraccionario

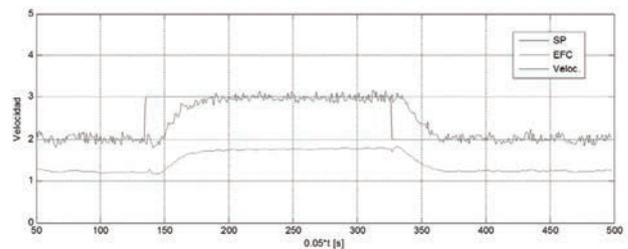


Figura 11. Respuesta con el controlador PID fraccionario Continuo equivalente

La figura 12 corresponde a la respuesta del sistema con el controlador PID fraccionario en su forma equivalente de potencias en z enteras la cual se obtuvo mediante el comando *nipid* y la opción *tustin*. La respuesta no presenta sobreimpulso tiene error cero en estado estable, menor tiempo de establecimiento que los casos anteriores y el trabajo de la ley de control sobre el elemento final de control es suave y sin oscilaciones considerables.

La figura 13 muestra el comportamiento del controlador PI fraccionario diseñado mediante ajuste por tablas. Se observa una respuesta sin sobreimpulsos, con error de estado estable igual a cero y tiempo de establecimiento pequeño. La acción sobre el elemento final de control es suave.

La figura 14 corresponde a la respuesta con el controlador PI fraccionario continuo equivalente. Esta respuesta presenta características similares a las del PI fraccionario original, aunque se observa una mayor sensibilidad en la respuesta que se traduce en pequeñas oscilaciones como se aprecia en la figura.

La figura 15 corresponde a la respuesta del sistema con el controlador PI fraccionario en su forma equivalente de potencias en z enteras la cual se obtuvo mediante el comando *nipid* y la opción *tustin*. El sistema presenta muy buena velocidad de respuesta, no presenta sobreimpulso tiene error cero en estado estable, y el trabajo de la ley de control sobre el elemento final de control es suave y sin oscilaciones considerables.

Resultados obtenidos con el PI y PID convencionales

La figura 16 muestra la respuesta del sistema con el controlador PI discreto convencional estimado utilizando el método del LGR y asignación de polos. Se puede observar que el sistema tiene muy buena velocidad de respuesta, sin sobreimpulso, con error de estado estable igual a cero y la acción de la ley de control sobre el elemento final es adecuada.

En la figura 17 se muestra la respuesta del sistema con el controlador PID Discreto Convencional estimado utilizando el método del LGR y asignación de polos. Se puede observar que el sistema tiene muy buena velocidad de respuesta, sin sobreimpulso, con error de estado estable igual a cero y la acción de la ley de control sobre el elemento final es adecuada.

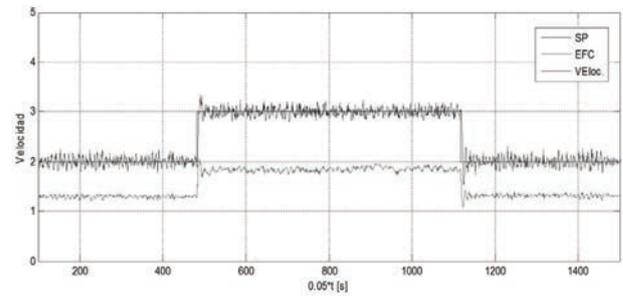


Fig. 12. Respuesta con el controlador PID fraccionario Discreto equivalente

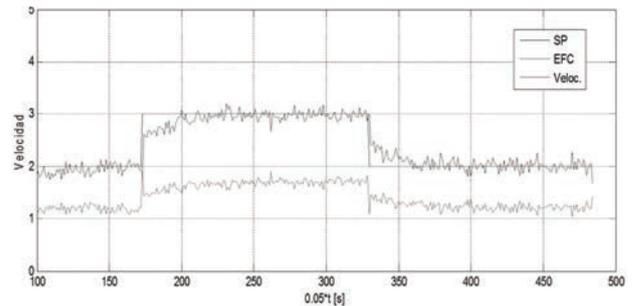


Fig. 13. Respuesta con el controlador PI fraccionario

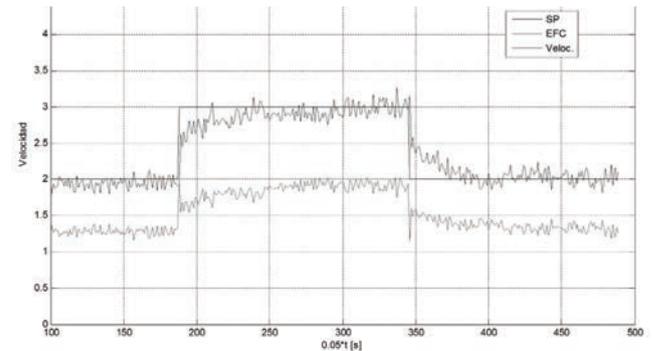


Fig. 14. Respuesta con el controlador PI fraccionario Continuo Equivalente

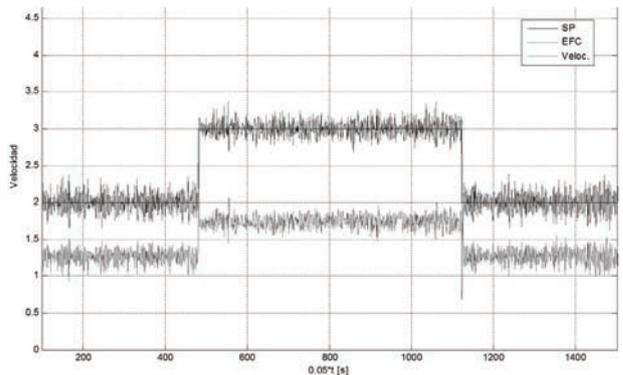


Fig. 15. Respuesta con el controlador PI fraccionario Discreto Equivalente

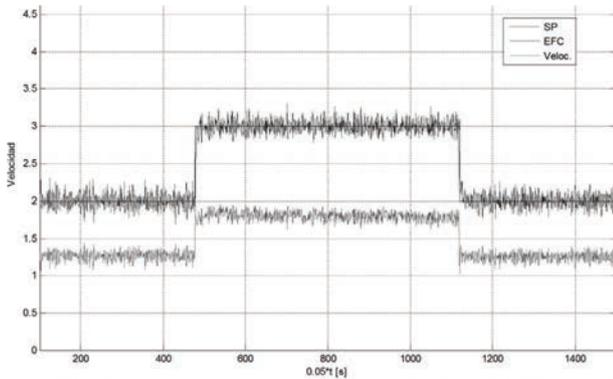


Fig.16. Respuesta con el controlador PI Discreto Convencional

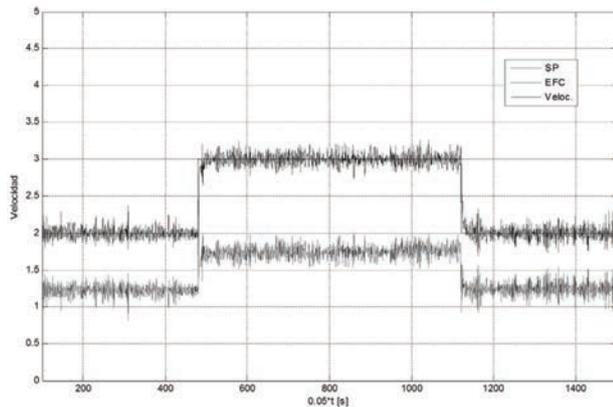


Fig. 17. Respuesta con el controlador PID Discreto Convencional

Comparación del desempeño en el proceso real

Para comparar el desempeño obtenido en la simulación con los diferentes controladores diseñados se utilizaron diferentes métricas con el fin de establecer suficientes criterios de análisis y

tener elementos cuantitativos y cualitativos para realizar la comparación.

Se analizan criterios de respuesta temporal como el tiempo de establecimiento, el máximo sobreimpulso y el error de estado estable, criterios de la integral del error como IAE, IAET, ICE e ICET y criterios de respuesta en frecuencia como el margen de ganancia y el margen de fase. En la tabla 1 se muestran los valores obtenidos con los diferentes criterios de comparación.

Los resultados indican que los controladores que presentan el mejor desempeño en cuanto a las métricas de respuesta temporal son el PIDE (PI fraccionario en su forma equivalente de potencias en z enteras) y el PI discreto diseñado mediante el método del LGR, mientras que en este aspecto los de peor desempeño son el PIDCE (PID fraccionario en su forma equivalente de potencias en S enteras) y el PICE (PI fraccionario en su forma equivalente de potencias en S enteras).

En cuanto a los criterios de error los de mejor desempeño son el PID y el PI discretos diseñados con el método del LGR.

En términos generales, se observó que el desempeño de los controladores discretos fue mejor que el de los controladores continuos, esto es un indicativo de la versatilidad de los controladores digitales y de su eficiencia al momento de regular un proceso.

Tabla 1. Comparación del desempeño de los controladores implementados

TIPO	IAE	IAET	ICE	ICET	t_s	$M_p\%$	e_{ss}	MG	ϕ_{PM}
$PI^\lambda D^\mu$	0.8909	0.6760	0.4820	0.2518	2	0	0	6.2	62.2
PIDDE	0.1659	0.0343	0.0806	0.0072	0.5	0	0	4.5	22.2
PIDCE	0.7147	0.3423	0.4907	0.1441	2.5	0	0	13.9	73.6
PIDLGR	0.0479	0.0013	0.0336	0.0005	0.2	2	0	11.1	62.0
PI^λ	0.0950	0.0138	0.0462	0.0020	0.2	20	0	5.3	54.0
PIDE	0.0873	0.0082	0.0459	0.0019	0.5	55	0	5.1	23.6
PICE	0.0820	0.0076	0.0422	0.0015	0.3	30	0	7.7	36.7
PILGR	0.0469	0.0011	0.0347	0.0005	0.2	0	0	12	71.0

$PI^\lambda D^\mu$: Controlador PID fraccionario.
 PIDDE: Equivalente discreto PID fraccionario.
 PIDCE: Equivalente continuo PID fraccionario.
 PIDLGR: Controlador PID convencional
 PI^λ : Controlador PI fraccionario.
 PIDE: Equivalente discreto PI fraccionario.
 PICE: Equivalente continuo PI fraccionario.
 PILGR : Controlador PI convencional.

Algunos autores como Ortiz y Cardona (2009) quienes diseñaron un control PID tipo fraccional para la posición del cabezal de una unidad de CD utilizado en el desarrollo de un microscopio óptico motorizado y Barbosa (2011) que compara el desempeño de un controlador PID adaptativo y de un controlador PID fraccionario para controlar la posición del vástago de un cilindro lograron un sistema de respuesta rápida, sin sobreimpulso y con una precisión aceptable, resultados comparables con los obtenidos en el presente trabajo.

5 CONCLUSIONES

- Se aplicaron las bases teóricas del cálculo fraccionario y del diseño de controladores PID fraccionarios y se demostró que estos controladores son una buena opción en el control de procesos.
- Se implementaron los algoritmos para el control PID fraccionario y PID convencional de acuerdo a la metodología de estudio establecida, arrojando resultados satisfactorios en el comportamiento del sistema.
- Con cualquiera de las estrategias de control aplicadas se obtiene un buen seguimiento de referencia de la velocidad del motor, con un comportamiento estable y operando de forma adecuada para una condición de operación particular. Con los controladores lineales PI clásico y PI fraccionario resulta factible conseguir respuestas muy buenas en términos de velocidad de respuesta, tiempo de establecimiento y máximo sobreimpulso.
- El PI fraccionario presenta ventajas frente al PI clásico, pues posee un grado de libertad más que el primero, permitiendo así ajustar a voluntad aún más características de la respuesta.
- Se aplicaron métodos de sintonía generales en el dominio de la frecuencia para los reguladores de orden fraccionario que garantizan el cumplimiento de las especificaciones de respuesta en frecuencia: frecuencia de cruce de ganancia (velocidad del sistema) y margen de fase (sobreimpulso).
- Se han tenido en cuenta en el diseño de controladores fraccionarios otros parámetros que mejoran el desempeño de los controladores como son la sensibilidad ante ruidos y perturbaciones cosa que no se puede hacer en la sintonía de los controladores PID convencionales.
- Se implementaron los sistemas de control PID fraccionario y PID convencional en el sistema real,

ambos controladores arrojaron un resultado satisfactorio en control de la salida de generación de energía. Al comparar el desempeño del controlador fraccionario con el control PID convencional por el método del lugar de las raíces se concluyó que ambos presentan un buen desempeño.

6 AGRADECIMIENTOS

Nuestro agradecimiento al Politécnico Colombiano Jaime Isaza Cadavid por el acompañamiento y suministros de implementos que permitieron llevar a cabo este trabajo.

7 REFERENCIAS

- [1] Barbosa, L. *Control fraccionario y adaptativo de un cilindro neumático*. Proyecto fin de carrera. Escuela Politécnica Superior de la Universidad Carlos III de Madrid. 2011.
- [2] Chen (2007). *Practical Tuning Rule Development for Fractional Order Proportional and Integral Controllers*. Journal of computational and nonlinear dynamics. Las Vegas. 2007.
- [3] Franklin et al. *Feedback Control of Dynamic Systems*. 4ª Edición, New Jersey. Ed. Prentice Hall. 2002.
- [4] García. *Control Digital. Teoría y práctica*. Politécnico Colombiano JIC. Medellín. Tercera Edición. ISBN 978-958-44-8641-7. 2011.
- [5] Hernández, O. *Metodología para modelar y controlar un sistema de combustión utilizando cálculo fraccional*. Tesis. Instituto Tecnológico Metropolitano. Facultad de Ingenierías. Maestría en Automatización y Control Industrial Medellín. 2013.
- [6] Johnson, V. *Diseño e implementación de estrategias de control avanzado sobre plataforma Labview para un sistema de calentamiento inductivo de fluidos*. Tesis de grado. Universidad de Chile Facultad de Ciencias Físicas y Matemáticas Departamento de Ingeniería Eléctrica. Santiago De Chile. 2010.
- [7] Matlab. *The MathWorks. MATLAB Documentation*, 9 edition. 2010.
- [8] Matlab. *Optimization Toolbox™ User's Guide* © COPYRIGHT 2013 by The MathWorks, Inc. 2013.
- [9] Monje, C.A., Vinagre, B.M., Chen, Y.Q., Feliu, V., Lanusse, P., Sabatier, J., "Proposals for fractional $PI^\lambda D^\mu$ tuning", *FDA'04*. pp. 156-161, Bordeaux, France. 2004.

[10] Oldham, K.B. y J. Spanier. *The Fractional Calculus*. Academic Press, New York. 1974.

[11] Ortiz, P y Cardona, L. Control PID tipo fraccional para la posición del cabezal de una unidad de CD para aplicaciones en microscopia óptica. ITECKNE Vol. 9 Número 2 • ISSN 1692 – 1798. 2012.

[12] Podlubny, I. *Fractional-Order Systems and $PI^\lambda D^\mu$ Controllers*. IEEE Transactions on Automatic Control 44 N° 1 Pp 208-214. 1999.

[13] Romero, H.M. *Control Predictivo de Orden Fraccionario*. Tesis Doctoral. Universidad Nacional de Educación A Distancia. Madrid. 2010.

[14] Sauchelli, V. H, Laboret, S. *Cálculo Fraccional Aplicado a Control Automático*. Mecánica Computacional. Volúmen X XVI, pp3308-3327. Córdoba, Argentina. 2007.

[15] Saxena, et al. "On Fractional Kinetic Equations." <http://arxiv.org/abs/math.CA/0206240>. 2002.

[16] Valério, D. *Fractional robust system control*. PhD thesis. Instituto Superior Técnico, Lisboa (Portugal). 2011.

[17] Valério. *Ninteger v. 2.3 Fractional control toolbox for MatLab*. Universidad Técnica de Lisboa Instituto Superior Técnico. 2011.

[18] Von Borries, M. *Estudio y Simulación de Sistemas Adaptables Fraccionarios*. Tesis de Maestría. Universidad de Chile Facultad de Ciencias Físicas y Matemáticas. Departamento Ingeniería Eléctrica. Santiago de Chile. 2012.