

CONTROLADOR DE POSICIÓN DIFUSO DE UN SEGUIDOR SOLAR CON INFERENCIA BASADA EN PROGRAMACIÓN LINEAL

Ramírez Julio, Angulo Omar, López José
Instituto Tecnológico de Nogales
División de Estudios de Posgrado e Investigación
Av. Tecnológico 911, Nogales, Sonora, México
+52(631) 311 1881
e-mail: jramirez@depiitn.edu.mx

RESUMEN.

El presente artículo trata sobre una nueva metodología de control difuso. En este nuevo enfoque, el motor de inferencia que usualmente consiste en reglas *si-entonces*, es remplazado por un método de optimización por *Programación Lineal*. En este método, las reglas de inferencia son transformadas en un conjunto de ecuaciones e inecuaciones, facilitando así, el uso de la *Programación Lineal* en la inferencia difusa. Para la prueba de resultados se implementa el caso de estudio de un seguidor solar, donde el desempeño del controlador propuesto, se compara con el de un controlador PID clásico y el de un controlador PD difuso.

Palabras Clave: Controlador PD difuso, Inferencia tipo Mamdani, Controlador PID, Programación Lineal.

ABSTRACT.

This paper discusses a new fuzzy control methodology. In this new approach, the inference engine which usually consists of if-then rules, it is replaced by Linear Programming optimization method. In this method, the inference rules are transformed into a set equations and inequalities, thus facilitating the use of Linear Programming in the fuzzy inference. In order to test this approach, a solar tracker case study is implemented. Then, we compare the proposed controller performance with a classic PID controller and a fuzzy PD controller.

Keywords: Fuzzy PD controller, Mamdani type inference, PID controller, Linear Programming.

1. INTRODUCCIÓN

El controlador PID clásico utiliza un paradigma de control muy discutido en la literatura [1]. Por otro lado el controlador PD difuso [2] consiste en un sistema de inferencia que se basa en los conceptos de la teoría de conjuntos difusos, reglas y razonamiento difuso. Este cuenta con numerosas aplicaciones de éxito en una gran variedad de campos, tales como el control automático, clasificación de datos, análisis para la toma de decisiones, sistemas expertos, robótica y reconocimiento de patrones.

Esta investigación está fundada en un trabajo anterior [3] publicado por los autores. Tal artículo trata sobre la representación de un controlador difuso tipo *Mamdani* por medio de ecuaciones relacionales difusas. En lo referente a la optimización en la inferencia de un sistema de base de reglas difusas, se tienen los trabajos de Sala [4] y Ramírez [5]. Aunque en ellos la base de reglas difusas era para la inferencia en el diagnóstico de fallos en sistemas industriales. En [6] se convierte un sistema de base de reglas difusas a un sistema de ecuaciones e inecuaciones para su validación pero no para la optimización en la inferencia. Otros como [7] usan lógica

binaria en lugar de ecuaciones e inecuaciones, aunque el planteamiento es similar. En [8] se basan en relaciones difusas para la inferencia pero no optimizan con Programación Lineal. En enfoques como [9,10] usan relaciones difusas para resolver problemas de Programación Lineal pero no para optimización en la inferencia.

La estructura de este artículo es como sigue: en la sección 2, se exponen los conceptos básicos en los que se basa nuestra propuesta. En la sección 3, se definen los principios de la inferencia difusa por optimización lineal. En la sección 4, se aplica la presente metodología a través de un caso de estudio. En la sección 5, se analizan los resultados obtenidos por los tres tipos de controladores probados en el seguidor solar. Por último, en la sección 6 tenemos las conclusiones.

2. DEFINICIONES BÁSICAS

En la teoría de control el controlador PID ha venido a ser uno de los controladores más usados en sistemas lineales. Para el control de sistemas no lineales el controlador PD difuso tiene bastante aceptación. En este trabajo se introduce un nuevo concepto, un controlador difuso cuya inferencia se implementa por medio de Programación Lineal en lugar de las tradicionales reglas *si-entonces*.

2.1. Controlador PID

Un PID (proporcional integral derivativo), es un mecanismo de control por realimentación, que calcula la desviación o error entre un valor medido y el valor que se quiere obtener, para aplicar una acción correctora que ajuste el proceso.

El algoritmo de cálculo del control PID se da en tres parámetros distintos: el proporcional, el integral, y el derivativo. El valor proporcional determina la reacción del error actual. El integral genera una corrección proporcional a la integral del error, esto nos asegura que aplicando un esfuerzo de control suficiente, el error de seguimiento se reduce a cero. El derivativo determina la reacción del tiempo en el que el error se produce, lo que genera un efecto predictivo. La acción de control proporcional, integral y derivativa se expresa según la ecuación (1).

$$G_c = K_P \cdot e + K_I \cdot \int e \, dt + K_D \cdot \frac{de}{dt} \quad (1)$$

2.2. Controlador PID difuso

Dado un controlador PD convencional, su ecuación está dada por la expresión (2).

$$G_c = K_P \cdot e + K_D \cdot \Delta e \quad (2)$$

donde K_P y K_D son los coeficientes de las ganancias proporcional y diferencial. Entonces un controlador difuso actuando como un PD consiste de reglas. La descripción simbólica de cada regla se enuncia como sigue:

si $e(k)$ es <símbolo de propiedad> y $\Delta e(k)$ es <símbolo de propiedad> **entonces** $u(k)$ es <símbolo de propiedad>.

En la anterior construcción, <símbolo de propiedad> es el nombre simbólico de un valor lingüístico (e.g., PG, i.e., Positivo Grande).

2.3 Programación Lineal

En un programa lineal general lo que se quiere es encontrar un vector $x \in \mathbb{R}^n$, maximizando (o minimizando) el valor de una función lineal dada entre todos los vectores $x \in \mathbb{R}^n$ que satisfacen a un sistema de ecuaciones lineales y desigualdades dado. La función lineal a ser maximizada o minimizada, se denomina la “función objetivo” y tiene la forma $c^T x = c_1 x_1 + \dots + c_n x_n$, donde $c \in \mathbb{R}^n$ es un hecho dado. Las ecuaciones lineales y las desigualdades en el programa lineal son llamadas “restricciones”. Es común denotar el número de restricciones por m . Un programa lineal se escribe frecuentemente usando matrices y vectores, en una manera similar a la notación $Ax = b$ para un sistema de ecuaciones lineales en álgebra lineal.

Minimizar una función objetivo $c^T x$ es equivalente a maximizar $-c^T x$, y por tanto, siempre podemos pasar a un problema de minimización. Después de las modificaciones cada programa lineal puede ser expresado como sigue:

“maximizar los valores de $c^T x$, entre todos los vectores $x \in \mathbb{R}^n$ satisfaciendo $Ax = b$ ”

donde A esta dado por la matriz real $m \times n$ y $c \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ son los vectores dados. Aquí la relación \leq vale para dos vectores de la misma longitud si y sólo si, se mantiene componente a componente. Cualquier vector $x \in \mathbb{R}^n$ satisfaciendo todas las restricciones de un determinado programa lineal, es una “solución factible”. Cada $x \in \mathbb{R}^n$ que da el máximo valor posible de $c^T x$ entre todos los x factibles, es llamada una “solución óptima”.

3. INFERENCIA DIFUSA POR OPTIMIZACIÓN LINEAL

Las reglas “**si** - **entonces**” pueden asumir la forma según la expresión (3):

si x es A **entonces** y es B (3)

donde A y B son valores lingüísticos definidos por conjuntos difusos en el universo de discurso X y Y , respectivamente. Es de común uso llamar a la expresión “ x es A ” **antecedente**, mientras que a la composición “ y es B ” **consecuente**. En este trabajo el universo de discurso X se refiere a la señal de error E del sistema y el universo de discurso Y se refiere a la señal de control M (ver Figura 4).

Una ecuación imagen de una base de reglas difusas puede formarse a partir de la *ecuación de implicación*, lo que está de

acuerdo con los requerimientos de generalización de la lógica binaria. Se usará para representar la operación de implicación la suma acotada de Lukasiewicz en donde “ $A \rightarrow B$ es *verdad*” equivale a “ $\min(1, 1-A+B) = 1$ ”, la cual es más conveniente que otros enfoques. Lo anterior es equivalente a una simple inecuación, que será usada como la definición básica para la implicación.

3.1 Implicación

La desigualdad equivalente que remplazará la implicación difusa “ $A \rightarrow B$ es *verdad*” es definida según la expresión (4):

$$A \leq B. \quad (4)$$

3.2 Doble implicación

Esto es “ $A \leftrightarrow B$ es *verdad*,” que representa $A \leq B$ y $B \leq A$ será equivalente a la expresión (5):

$$A = B \quad (5)$$

de esta manera, la inferencia es transformada en la solución de un conjunto de igualdades y desigualdades.

3.3 Definición de variables

En esta propuesta, el sistema de inferencia difuso será tratado desde el punto de vista de los controladores difusos tipo *Mamdani*. Aunque el enfoque aquí discutido, también puede ser aplicado a los controladores tipo *Sugeno*.

El sistema de inferencia difuso al que nos referiremos se encuentra descrito en el diagrama a bloques de la Figura 1, donde desde arriba hacia abajo X es la señal de entrada, esto es, la señal de error (E). X_j corresponde a el antecedente de la regla de inferencia “**si** X es A_j ” y Y_i equivale al consecuente de la misma regla, esto es: “**entonces** Y es B_i ”. También nos podemos referir a Y como la señal de control (M).

La salida del bloque X_j equivale a $\mu_{A_j}(x)$. La salida del bloque Y_i equivale a $\mu_{B_i}(y)$ donde $B' \equiv B$ es recortado por $\mu_{A_j}(x)$. El *agregador* es donde se lleva a cabo la operación de implicación difusa tipo *Mamdani* que conlleva la composición max-min. A la salida del sistema tenemos el *desdifusificador* que en esta aplicación es del tipo centroide (COA) y nos genera una salida nítida (*crisp*).

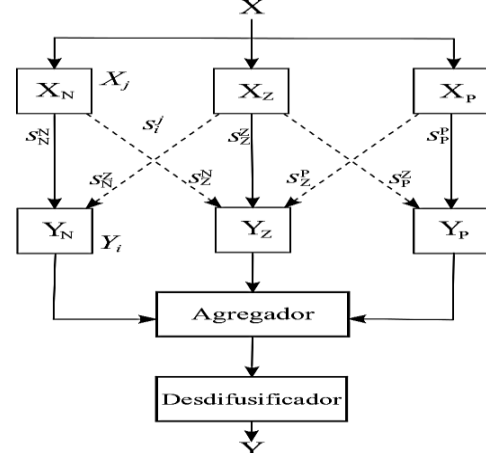


Figura 1. Sistema de inferencia de asociaciones difusas.

Ahora se hace referencia a las asociaciones difusas existentes entre los bloques X_i y Y_j que constituyen el tema de discusión principal de esta aportación. La variable auxiliar S_i^j denota la relación existente entre los antecedentes y los consecuentes de las reglas. En el caso concreto observado en la Figura 1, la flecha continua de S_N^N corresponde a una asociación directa (igualdad). Esto es, una relación de equivalencia entre las funciones de pertenencia asociadas directamente (donde $\mu_{A_j}(x) = \mu_{B_i}(y)$ para cualquier valor de x). Las flechas de puntos denotan una relación difusa “incierto” (desigualdad) entre los antecedentes y consecuentes. En esta situación tendríamos a S_N^Z , S_Z^N , S_Z^P y S_P^Z .

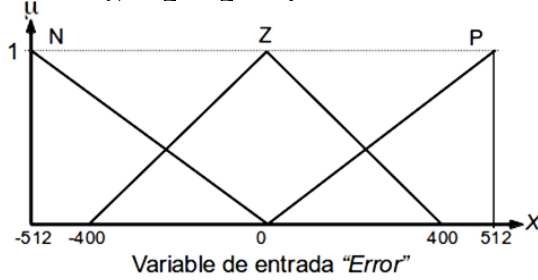


Figura 2. Funciones de pertenencia para la variable de entrada “X” esto es el “Error”.

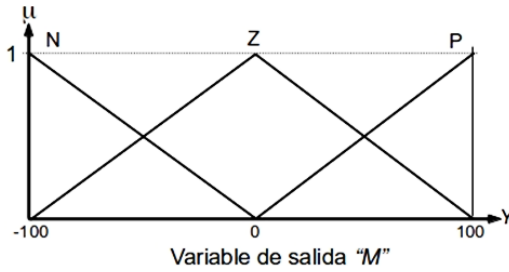


Figura 3. Funciones de pertenencia para la variable de salida “Y”, esto es la salida del actuador “M”.

Las Figuras 2 y 3 son las funciones de pertenencia usadas en nuestro caso de estudio que se verá en la sección 4. En este contexto, S_N^Z se refiere a una relación difusa incierta entre la función de pertenencia N de Y (ver Figura 3) y la función de pertenencia Z de X (ver Figura 2). De igual forma S_Z^N se refiere a la relación difusa incierta entre la función de pertenencia Z de Y (ver Figura 3) y la función de pertenencia N de X (ver Figura 2). Para las variables S_Z^P y S_P^Z aplica el mismo principio. La Optimización Lineal determinará el valor de estas relaciones en el intervalo (0,1) dando un valor gradual similar a la inferencia difusa.

3.4 Asociación antecedente-consecuente

Dadas las funciones de pertenencia del error *Negativo*, *Cero*, *Positivo*, (N , Z y P respectivamente) las reglas tratan de modelar la función identidad $X = Y$ y serán:

$$\begin{aligned} X \text{ Negativo} &\leftrightarrow Y \text{ Negativo}, X \text{ Cero} \leftrightarrow Y \text{ Cero}, \\ X \text{ Positivo} &\leftrightarrow Y \text{ Positivo}. \end{aligned}$$

Las $X_j(x)$ identifican a las funciones de pertenencia de los antecedentes A_j tal que $x_j(x) = \mu_{A_j}(x)$, de igual forma se tiene que $Y_i(y) = \mu_{B_i}(y)$. La asociación o relación entre un antecedente X_j y un consecuente Y_i se expresará por una “variable auxiliar de asociación”, denotada por S_i^j (ver Figura 1) que observa los siguientes condicionamientos:

Asociaciones ciertas. Si el antecedente X_j está directamente relacionado con el consecuente Y_i se trata de una asociación directa entre las funciones de pertenencia involucradas de tal manera que satisface el condicionamiento de igualdad, según se muestra en la ecuación (6).

$$S_i^j = Y_i \quad (6)$$

En nuestro caso de estudio, se establece que las asociaciones ciertas con respecto a la variable S_i^j son las ecuaciones (7).

$$S_N^N = Y_N, \quad S_Z^Z = Y_Z, \quad S_P^P = Y_P \quad (7)$$

Asociación inexistente. Si la asociación entre el antecedente X_j y el consecuente Y_i es inexistente, entonces la variable auxiliar tiene el valor de cero (8).

$$S_i^j = 0 \quad (8)$$

Asociaciones inciertas. Si la asociación entre los antecedentes X_j y los consecuentes Y_i es “incierto” (esto es, puede ser una igualdad o una desigualdad dependiendo del valor de X), se agregará sobre el condicionamiento mostrado en la expresión (9).

$$S_i^j \leq Y_i \quad (9)$$

Las asociaciones inciertas tienen lugar en las partes graduales difusas de las funciones de pertenencia de X_j y es incierta porque la relación de S_i^j con respecto a Y_i depende de los valores de X en el intervalo de gradualidad donde se traslapan las funciones de pertenencia. Si no se traslapan no existirá una relación difusa entre las premisas de las reglas.

Las expresiones (10) son las ecuaciones e inecuaciones de asociaciones inciertas usadas en nuestro caso de estudio.

$$S_N^Z \leq Y_N, \quad S_Z^N \leq Y_Z, \quad S_Z^P \leq Y_Z, \quad S_P^Z \leq Y_P \quad (10)$$

Ecuaciones de superposición. Una vez que se han fijado las restricciones o condicionamientos en los coeficientes de relación S_i^j , el estado inferido de la relación de los antecedentes X_j con respecto a los coeficientes S_i^j será dado por la disyunción [5] sobre todos los posibles consecuentes con los que se relaciona (un dado nodo X_j), que se podría expresar de la forma vista en la expresión (11).

$$X_j = \min\left(1, \sum_{i=1}^k S_i^j\right) \quad (11)$$

Sin embargo, (11) no es lineal. Con el fin de preservar la linealidad, se agregarán las siguientes desigualdades (12) a la base de conocimiento en lugar de (11):

$$\begin{aligned} S_i^j &\leq X_j \quad \forall i \\ X_j &\leq \sum_i S_i^j \end{aligned} \quad (12)$$

La primera ecuación indica que el valor difuso a la salida del antecedente X_j es más grande o igual que la salida del consecuente con el que se relaciona. La segunda ecuación indica que al menos una relación con los consecuentes debe ser verdadera para que la salida X_j no sea cero. En este contexto, la interpretación de $S_i^j = 1$ es “ Y_i relaciona con toda certeza a X_j ”. Las ecuaciones (12) fueron propuestas primero en [11] y adaptadas para su aplicación aquí.

Las ecuaciones de superposición, aseveran la verdad de las entradas X_j con respecto a S_i^j y la posible existencia de varias gradualidades de verdad de X_j con respecto a los consecuentes con los que se relaciona. Las siguientes expresiones, son las ecuaciones de superposición de nuestro caso de estudio.

$$\begin{aligned} S_N^N &\leq X_N, S_Z^N \leq X_N, S_N^N + S_Z^N \geq X_N, \\ S_Z^Z &\leq X_Z, S_Z^Z \leq X_Z, S_P^Z \leq X_Z, S_N^Z + S_Z^Z + S_P^Z \geq X_Z, \\ S_2^3 &\leq X_3, S_3^3 \leq X_3, S_2^3 + S_3^3 \geq X_3 \end{aligned}$$

3.5 Inferencia como problema de satisfacción de restricciones

Dada la señal de control X y las restricciones especificadas en (6)-(12), el problema de inferencia en el controlador difuso puede formularse como un problema de satisfacción de restricciones. Como todas las restricciones argumentadas son lineales con respecto a las variables desconocidas, la Programación Lineal puede usarse como una herramienta de solución. En este trabajo se usó para este efecto el software llamado MATLAB. La teoría elemental de Programación Lineal, nos muestra que un conjunto de variables de decisión factibles, serán un polítopo convexo.

Optimización por Programación Lineal. Enumerar los posibles vértices polítopos es muy impráctico. Con el fin de convertir el problema a una optimización de Programación Lineal con solución única, se propone la implementación de la siguiente técnica:

Primero, agregamos la siguiente ecuación (13) a la base de conocimiento

$$X_j = x_j + \epsilon_j - \gamma_j, \text{ para } \epsilon_j \geq 0, \gamma_j \geq 0, \quad (13)$$

donde ϵ_j y γ_j son variables auxiliares y $x_j \approx X_j$ (entonces ahora $x_j = \mu_{A_j}(x)$). De esta manera, es factible tener una solución única.

Segundo, se construye la función objetivo J a ser “minimizada”. El índice J consiste de tres términos:

- La suma de todas las variables ciertas con una posibilidad de certeza π_c por lo que se ponderan con $\omega_c \rightarrow 0$, esto es $J_1 = [\omega_c S_N^N + \dots]$.
- La suma de los coeficientes de asociación S_i^j para los cuales $i \neq j$. Esto es, según se muestra en la ecuación (14), todos los coeficientes de relaciones difusas inciertas.

$$J_2 = \sum_{i \neq j} \omega_i^j S_i^j \quad (14)$$

donde ω_i^j debe ser aprendida de tal forma que pueda seguir a la función de control $f(x)$.

- La suma de las variables auxiliares ponderadas $J_3 = \sum (\omega_j \epsilon_j + \omega_j \gamma_j)$. Con el propósito de lograr que $x_j \approx X_j$,

las variables ϵ_j y γ_j deben ser ponderadas en forma significativa ($\omega_j \rightarrow \infty$).

En el marco de trabajo de optimización, la más pequeña J es la más “posible” que debe ser considerada, de acuerdo a las ideas en [12]. De esta forma, la posibilidad marginal de alguna variable puede ser denotada por la siguiente función:

$$\pi_i^{marg}(x) = e^{-\min J_i(x)} \quad (15)$$

En este contexto, todas las variables ciertas (sin incluir las de (13)) tienen una $\pi = 1$, por lo que si se les quiere incluir en el índice de costes J , deben ser ponderadas con un valor muy pequeño ($\omega_i \rightarrow 0$). Por lo anterior las únicas variables cuya ponderación debe ser aprendida, son las variables de las que no tenemos una idea clara como debemos ponderar. Esto es, los coeficientes de asociación difusos inciertos (14).

Función objetivo J . En este índice se incluyen todas las variables (excepto las x_j que son las entradas a nuestro sistema), cada una de las cuales es ponderada con un peso.

Si no se quiere tener un sistema muy optimizado basta con proponer una $\pi = 0.999$ a todas las variables ciertas ($\omega = 0.001$ para facilitar la optimización por Programación Lineal) y una $\pi = 0.900$ a todas las variables inciertas J_2 . Con estos valores del índice J en la mayoría de los sistemas sencillos la salida Y seguirá a la función de control $f(x)$. En caso de que se desee tener un índice optimizado se propone en [3] un algoritmo que aprende la función de control $f(x)$.

4. CASO DE ESTUDIO: CONTROLADOR DE UN SEGUIDOR SOLAR

El seguidor solar está compuesto principalmente por ocho elementos: un sensor de posición óptico, una tarjeta de adquisición de datos (1018 PhidgetInterfaceKit 8/8/8), un “driver” (Microcontrolador HA) para motor, un motor (36JX30K139G/3863-1230 139:1 “Planetary Gearbox DC Motor”), una computadora, una batería, una estructura mecánica y un panel solar. Aunque en realidad la computadora y los primeros cuatro elementos son quienes participan directamente en el control. En la Figura 4, se muestra un diagrama a bloques del seguidor solar. El sistema completamente integrado se puede ver en la Figura 5.

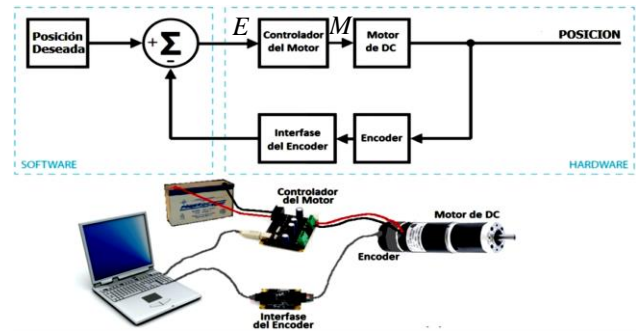


Figura 4. Diagrama a bloques del seguidor solar.

4.1. Controlador difuso

El controlador difuso clásico es del tipo PD (Proporcional Derivativo). En estos controladores se toman en cuenta dos

señales de entrada: el Error (E) y la derivada del Error ($\frac{dE}{dt}$). En base a estas entradas es que tenemos las acciones de control que se muestran en el diagrama a bloques de la Figura 6.



Figura 5. Sistema completamente integrado.

De esta forma, las variables de entrada al controlador son: Error y derivada del Error. La variable de salida (M) es la acción de control.

Los valores lingüísticos para las variables de entrada y de salida son: Negativo (N), Cero (Z) y Positivo (P).

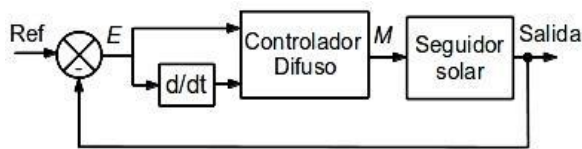


Figura 6. Controlador difuso del seguidor solar.

La tarjeta de adquisición de datos Phidget que es usada para leer los datos del sensor de posición óptico, usa un convertidor analógico digital de 10 bits, por lo que el valor digital del error oscilará entre ± 512 . Por lo anterior los valores paramétricos (a , b , c) de las funciones de pertenencia triangular (ver Figura 7 y 2) de la señal de Error son: para Negativo (-512, -512, 0), para Cero (-400, 0, 400) y para Positivo (0, 512, 512).

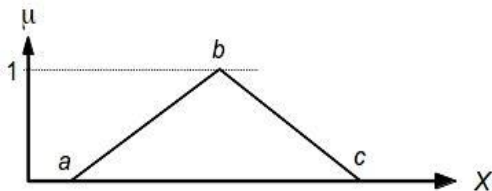


Figura 7. Función de pertenencia triangular.

Para la señal derivada del Error los parámetros son: para Negativo (-512, -512, 0), para Cero (-512, 0, 512) y para Positivo (0, 512, 512). Para la variable de salida M del controlador los valores lingüísticos serán: para Negativo (-100, -100, 0), para Cero (-100, 0, 100) y para Positivo (0, 100, 100) (ver Figura 7 y 3). Como se observa los parámetros para las funciones de pertenencia de salida oscilan entre -100 y 100. Esto se debe a que el actuador es una tarjeta Phidgets (tipo puente H) que genera una señal por modulación del

ancho de pulso. El 100% de ancho de pulso, aplica un voltaje máximo de 15 voltios (al motor que se encarga de la posición del panel solar,) en un sentido. El -100% del ancho de pulso aplica un máximo de -15 voltios, que invierte el sentido de movimiento del motor.

De acuerdo con las señales de Error y derivada del Error, es que se decide que valores tendrá la variable de salida M . Este conocimiento es experimental y se expresa en una serie de reglas de inferencia **si – entonces**. En nuestro caso se implementaron nueve reglas organizadas en una FAM (*Fuzzy Associative Memory*) que se muestran en la Tabla 1.

Tabla 1. FAM del controlador difuso.

$E \backslash \frac{dE}{dt}$	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

4.2. Controlador difuso con Inferencia por Programación Lineal

Después de haber configurado nuestro controlador difuso clásico, ahora pasaremos a definir el controlador difuso con inferencia por Programación Lineal. Debido a que nuestra aportación aún se encuentra en una etapa inicial, y de acuerdo con la teoría vista en la sección 3, solo seremos capaces de hacer la inferencia cuando $X = Y$. Esto significa que se consideró para esta inferencia, la señal de Error (E), como única entrada y la acción de control (M), como salida. Esto implica que $E = X$ y Y produce M . Nuestro controlador genera la acción de control Proporcional, que será nuestra única acción de control. Así pues, nuestras tres reglas de inferencia por Programación Lineal serán:

si X es N **entonces** Y es N

si X es Z **entonces** Y es Z

si X es P **entonces** Y es P

El diseño del controlador se encuentra definido en la sección 3, en donde de las asociaciones ciertas (7) tenemos 3 ecuaciones. De las asociaciones inciertas (10) tenemos 4 inequaciones. De las ecuaciones de superposición tenemos 10 inequaciones y de las variables auxiliares (13) tenemos 3 ecuaciones. Para 3 reglas como las de nuestro controlador difuso con inferencia por Programación Lineal resultan un total de 20 expresiones entre inequaciones y ecuaciones. Sin embargo, lo que determina el tamaño de las matrices usadas en Programación Lineal, es el número de variables contenidas en el total de expresiones. En la Figura 1, podemos contabilizar el número mínimo de 13 variables de nuestro controlador, más las 6 variables auxiliares (13), tendremos un total de 19. Así que el tamaño mínimo de nuestras matrices en la definición del problema de Programación Lineal es de 19.

La implementación de este controlador se hizo en MATLAB. El comando utilizado es *linprog*. El problema a resolver en el diseño de este controlador, es de minimización de la función objetivo J . El modo de uso del comando *linprog* es: $Y = \text{linprog}(J, A, b, Aeq, beq, LB, UB)$, donde J es la función objetivo, A es la matriz de desigualdad, b es un vector que contiene el

segundo miembro de las desigualdades, A_{eq} es la matriz de ecuaciones, b_{eq} es un vector del segundo miembro de las ecuaciones, LB y UB son respectivamente, los límites inferior y superior de las variables del problema. La salida Y contiene un vector con los respectivos valores de Y_N , Y_Z y Y_P , *singletons* usados por el agregador para luego generar la salida M del desfudificador.

En el índice J se representan todas las variables (excepto las entradas x_i , que son las entradas al sistema de inferencia difuso), cada una de las cuales es ponderada con un peso. Se sugiere [3] que este peso sea pequeño en el orden de $\omega = 0.001$ para todas las variables ciertas y mayor que este valor, para todas las variables inciertas.

La implementación de este sistema se hizo en un computador bajo el sistema operativo Windows 7. El software y sus interfaces de usuario se desarrollaron en C#. La comunicación con las tarjetas se hizo a través del puerto USB utilizando *dlls* provistos por la compañía Phidgets. Dado que el software se desarrolló en C#, el programa de inferencia difusa por Programación Lineal desarrollado en MATLAB, se convirtió en una librería de enlace dinámico (*dll*). Al hacer esto, la librería se usa como una clase en C#. Esta clase contiene una función a la que le pasamos la señal de error (E) y nos regresa los valores Y_N , Y_Z y Y_P . Con estos valores se genera la señal M que pasamos al *driver* actuador del motor de CD.

Tanto el controlador difuso clásico como el controlador difuso con inferencia por Programación Lineal y el controlador PID, todos se implementaron en C#. El algoritmo del PID se digitalizó usando la técnica de integración por suma trapezoidal y derivación por la diferencia entre dos puntos.

4. ANÁLISIS DE RESULTADOS

Una vez implementado el software e integrado el hardware del seguidor solar, se hicieron pruebas comparativas entre los tres controladores. En la Figura 8 tenemos la respuesta comparativa de los tres tipos de controladores estudiados. Aunque el comportamiento de estas respuestas depende mucho del ajuste de parámetros, podemos observar que el controlador PID tiene buen comportamiento con un pequeño sobre paso y rápida estabilización.

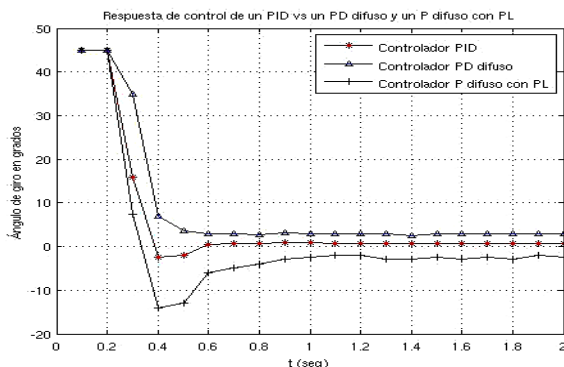


Figura 8. Respuesta de un controlador PID, PD difuso y P difuso con inferencia basada en Programación Lineal.

El controlador PD difuso no tiene sobrepaso y tiene una rápida estabilización, (aunque observa mayor error que el PID). Este error se pudo haber reducido afinando o

aumentando el número de funciones de pertenencia de salida, pero por compatibilidad con el controlador P difuso con inferencia por Programación Lineal, se implementaron tres funciones de pertenencia. Es evidente que el controlador con menor desempeño es el controlador difuso con inferencia por Programación Lineal. Sin embargo, este resultado es prometedor, pues creemos que con la expansión a una entrada de control más, que sería la señal derivada del error, aplicando la metodología aquí propuesta, tendría el mismo desempeño que el controlador difuso.

5. CONCLUSIONES

Podemos concluir, que aunque el PID tiene mejor desempeño, la ventaja de los controladores difusos es que se pueden aplicar a sistemas no lineales. El controlador propuesto en este trabajo, aunque no es el mejor, tiene un desempeño aceptable, además que por su naturaleza matemática sus reglas de inferencia pueden ser validadas para evitar las contradicciones y garantizar que todas las funciones de pertenencia sean mutuamente exclusivas y excluyentes.

6. REFERENCIAS

1. K. Ogata. *Modern Control Engineering*. Prentice Hall, 2002.
2. D. K. Chaturvedi. *Soft Computing*. Springer, 2008.
3. J. C. Ramírez. *Inference optimization approach in fuzzy inference systems*. IEEE Computer Society, CERMA 2008, 1:56-61, 2008.
4. A. Sala *et al.* *An optimization approach to fuzzy diagnosis: Oil analysis application*. Fuzzy System Conference, FUZZY-IEEE International, 1:1-6, 2007.
5. J. C. Ramírez. *Failure Diagnosis in Industrial System Based on Possibilistic and Fuzzy Reasoning*, Phd Tesis. Polytechnic University of Valencia, Spain, 2007.
6. A. Sala y P. Albertos. *Inference Error Minimisation: Fuzzy Modelling of Ambiguous Functions*. Fuzzy Sets and Systems. 1(121), 95-111, 2001.
7. A. F. Gobi y W. Pedrycz. *Fuzzy Modelling Through Logic Optimization*. International Journal of Approximate Reasoning. Elsevier, 1(45):488-510, 2007.
8. A. B. Rakityanskaya *et. al.* *Fuzzy Relational-Based Diagnosis*. Automation and Remote Control, Pleiades Publishing, 68(12):2198-2219, 2007.
9. Adel, Hatami-Marbini, y Madjid, Tavana. *An extension of the linear programming method with fuzzy parameters*. International Journal of Mathematics in Operational Research, Inderscience, 3(1):44-55, 2011.
10. A. Ghodousian y E. Khorram. *Fuzzy Linear Optimization in the Presence of the Fuzzy Relation Inequality Constraints with max-min Composition*. Information Sciences, Elsevier, 1(178):501-519, 2008.
11. A. Sala and Pedro Albertos. *Fuzzy systems evaluation: The inference error approach*. IEEE Transactions on Systems evaluation, Man, and Cybernetic part B: Cybernetics, 28(2):268-275, 1998.
12. D. Dubois, H. Fargier, and H. Prade. *Possibility theory in constrain satisfaction problem: handling priority, preference constrain and uncertainty*. Applied Intelligence, 6(4):287-309, 1996.