

Diferentes arquitecturas aplicadas a la implementación de un Controlador Difuso para el péndulo invertido

Nelson Acosta y Daniel Simonelli¹

INCA - Facultad de Ciencias Exactas

Universidad Nacional del Centro de la Provincia de Buenos Aires

TE.: +54-2293-432466 / (7000) Tandil - Argentina

E-mail: dsimonel@exa.unicen.edu.ar

Resumen

Se utilizó un péndulo como dispositivo controlado en el prototipo de un sistema de control de tiempo real empleando diferentes arquitecturas. Se diseñaron algoritmos de control para la evaluación como así también el hardware requerido para comprobar su comportamiento. En estas condiciones, se obtuvieron como métricas del desempeño de las cinco arquitecturas, el número de Inferencias Difusas por Segundo (FIPS) y el número de Reglas Difusas por Segundo (FRPS). Se analizaron dichos valores y se discuten futuras ampliaciones.

1. Introducción

Muchas de las aplicaciones de control industrial presentan un comportamiento no lineal y sus modelos correspondientes habitualmente se hallan repletos de incertidumbres lo que dificulta la materialización de controladores adecuados para tales aplicaciones. Con el propósito de solucionar los problemas en el diseño de controladores de sistemas no lineales, en las últimas dos décadas se han propuesto una buena cantidad de enfoques satisfactorios basados principalmente en lógica difusa [Lee01, Sch00]. Otras áreas de investigación con aplicación en diversos campos de la ingeniería, la física, la química o la biología tales como identificación, sincronización y control de sistemas dinámicos caóticos han aportado una nueva familia de problemas cuyas soluciones se basan en la aplicación de técnicas de inferencia difusa [Cha01, Che99].

Todas estas aplicaciones cuya solución se realiza aplicando sistemas difusos requieren de controladores difusos (FLC's) cada vez más veloces. Para satisfacer esta necesidad de reducir los tiempos de respuesta se trabaja con dos grandes enfoques:

- Optimización de los algoritmos. Se busca reducir el tiempo de cálculo de cada una de las etapas de cálculo difuso [Aco98, Chi92, Chu96, Kos94]: fusificación, inferencia y defusificación.
- Mejoramiento de las plataformas de cálculo. Trabajos de investigación y diversas materializaciones se agrupan en aplicaciones basadas en microcontroladores dedicados [Jag95, Wan96], software sobre plataformas genéricas [Jan95, Aco01] y circuitos a medida de la aplicación [Aco98, Aco00, Des97, Des98, Tog96, Wat90].

El presente trabajo describe y analiza la materialización de cinco diseños en software de FLC's sobre computadoras genéricas. Para comparar los diferentes enfoques de materializaciones se necesita de un problema práctico lo suficientemente *bien* definido como para no generar problemas inherentes al ejemplo y por lo tanto no imputables a las herramientas o a la metodología.

¹ - Becario de perfeccionamiento de la CICPBA (Comisión de Investigaciones Científicas de la Prov. de Buenos Aires).

En este sentido, la aplicación seleccionada es un controlador para mantener en equilibrio un “*péndulo invertido*”. Este controlador pertenece a la familia de sistemas clásicos con modelo matemático no lineal lo suficientemente complejo como para justificar un control que no se base en dicho modelo [Coo94, Wan96].

2. Descripción del sistema de control

El sistema de control considerado se muestra en la Fig. 1, y consiste del modelo matemático para el péndulo invertido. En el mencionado modelo, se lee periódicamente el ángulo α (por medio del Encoder) y la velocidad angular ν se calcula por diferencias entre lecturas consecutivas del ángulo ("Diferenciador"). Los valores del ángulo y velocidad (angular) se pasan como entradas al FLC, el cual calcula la fuerza de control f a aplicarse al péndulo invertido.

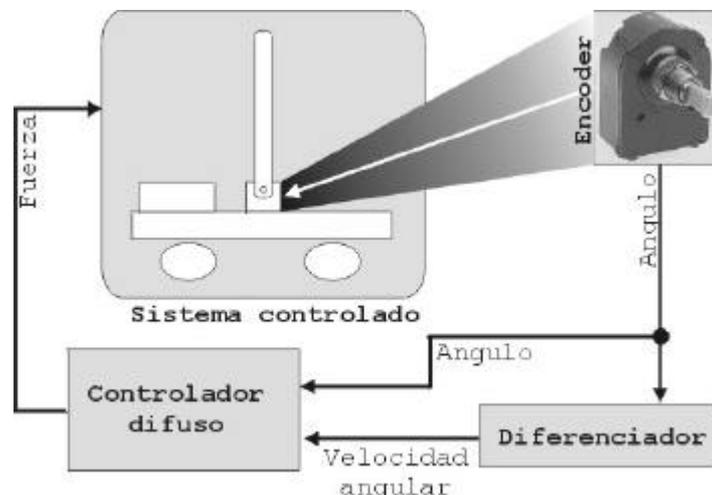


Fig. 1 – FLC del péndulo invertido

El FLC se especifica en alto nivel, las reglas definen las relaciones entre las variables lingüísticas difusas (o sustantivos difusos): Ángulo, Velocidad y Fuerza; y sin utilizar un modelo matemático complejo.

3. Definición del modelo del péndulo invertido

En la Fig. 2 se presenta el modelo del péndulo invertido. Físicamente, el péndulo consiste de una vara de longitud l que pivotea en un carro que se desplaza sobre el plano horizontal. Se considera una versión simplificada del modelo completo, en la cual se supone que el carro no tiene masa y que la masa de la viga m se concentra en su centro. Sobre m actúan dos fuerzas, gravedad (aceleración gravitacional g) y la fuerza de control externa f que es la responsable de estabilizar al sistema. Las otras variables son el ángulo de inclinación con respecto a la perpendicular α , la velocidad angular ν y la aceleración angular j .

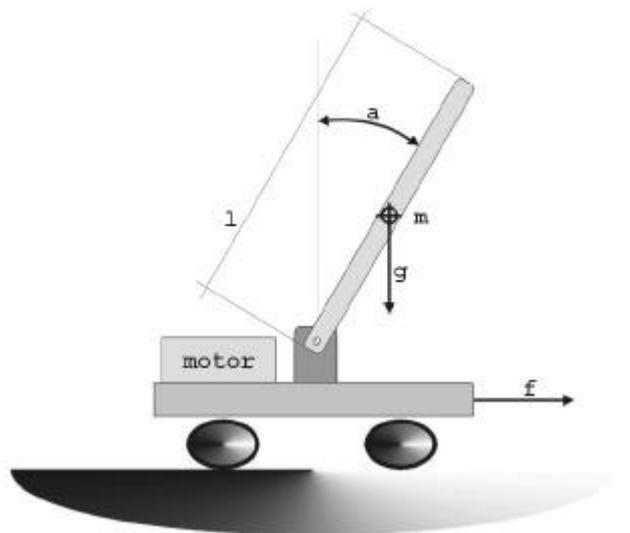


Fig. 2 – Sistema del péndulo invertido

En todo momento, el péndulo está gobernado por las siguientes ecuaciones [Syn95]:

$$da = v dt + 0,5 j dt^2 \quad (1)$$

$$dv = j dt \quad (2)$$

$$j = \frac{3 g}{2 l} \sin(a) + \frac{6}{m l} f \cos(a) \quad (3)$$

De la ecuación (3) se deducen los valores límites para la fuerza de control f necesaria para equiparar la aceleración gravitacional (para cierto ángulo a) y llevar al péndulo a la posición de equilibrio.

$$\left| \frac{6}{m l} f \cos(a) \right| > \left| \frac{3 g}{2 l} \sin(a) \right| \quad \text{ó} \quad |f| > \frac{g}{4} |\tan(a)|$$

$$|f| > 2.85 g, \text{ para cuando el ángulo } a = 85^\circ \text{ (cerca del peor caso, i.e. cuando } 90^\circ \text{)}. \quad (4)$$

4. Diseño del FLC

Se deben especificar las variables de entrada **Ángulo**, **Velocidad** y la variable de salida **Fuerza**. Las funciones de pertenencia difusas se concentran cerca del punto de equilibrio deseado para dotar de una mayor precisión al proceso de estabilización.

Para la variable de entrada “Ángulo” se especifica un dominio en el intervalo $[-90^\circ, +90^\circ]$ para los valores mínimo y máximo, respectivamente. Dicho rango se divide en 7 funciones de pertenencia con un 50% de superposición que se muestran con detalle en la Fig. 3.

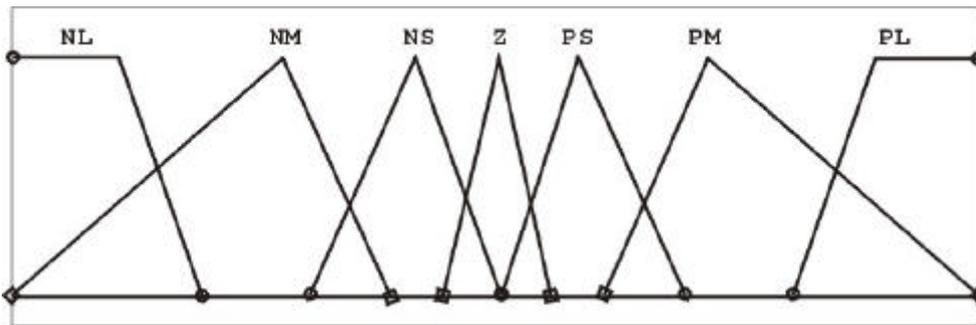


Fig. 3 – Ángulo del péndulo

Para la variable de entrada “Velocidad” se especifica un dominio en el intervalo $[-30, +30]$. Dicho rango se divide en 7 funciones de pertenencia con un 50% de superposición que se muestran con detalle en la Fig. 4, a saber, velocidad negativa grande (NL), negativa mediana (NM), negativa pequeña (NS), nula (Z), positiva pequeña (PS), positiva mediana (PM) y positiva grande (PL).

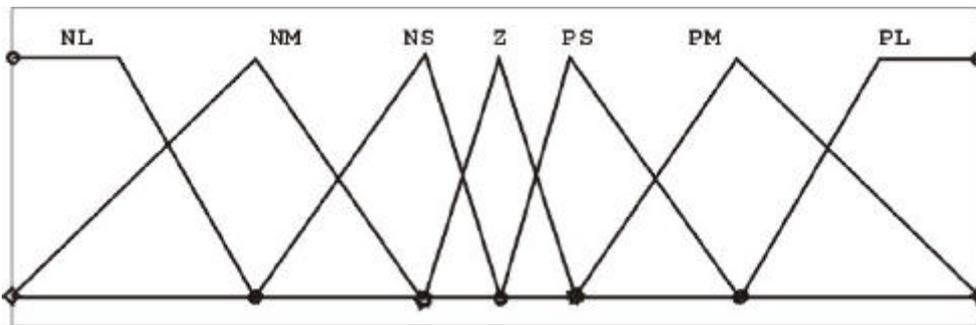


Fig. 4 – Velocidad angular difusa (VELOCIDAD)

Para la variable de entrada “Fuerza” se especifica un dominio en el intervalo $[-30, +30]$. Dicho rango se divide en 7 funciones de pertenencia con un 50% de superposición. Las funciones de pertenencia se muestran con detalle en la Fig. 5, y consisten de: ángulo negativo grande (NL), negativo mediano (NM), negativo pequeño (NS), nulo (Z), positivo pequeño (PS), positivo mediano (PM) y positivo grande (PL).

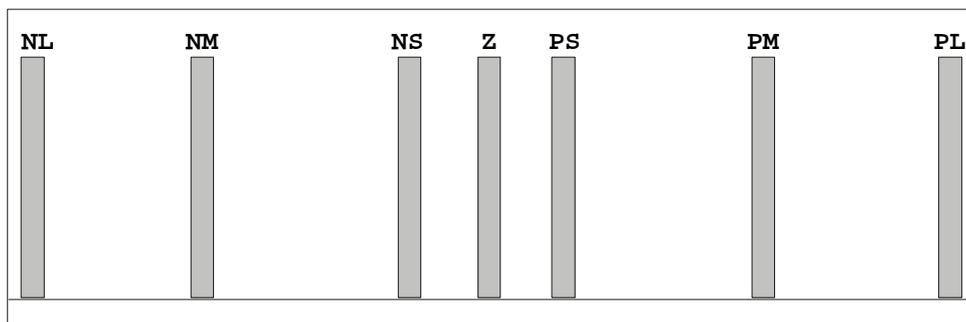


Fig. 5 – Singleton de la variable de salida Fuerza

Para las variables de entrada “Ángulo” y “Velocidad” se debe definir la relación sobre la variable de salida “Fuerza” en términos de reglas. Ésta última presenta siete funciones de pertenencia: negativa grande (NL), negativa mediana (NM), negativa pequeña (NS), nula (Z), positiva pequeña (PS), positiva mediana (PM) y positiva grande (PL). Basándose en el sentido común, intuición y experiencia se definen las guías para controlar al péndulo: "si el ángulo y la

velocidad son del mismo signo, entonces la fuerza debe ser de signo opuesto", con algunas variaciones menores cerca del punto de equilibrio. La Fig. 6 muestra la tabla de reglas.

		VELOCIDAD						
		NL	NM	NS	Z	PS	PM	PL
ÁNGULO	NL	PL	PL	PM	PM	PS	PS	Z
	NM	PL	PM	PM	PS	PS	Z	NS
	NS	PM	PM	PS	PS	Z	NS	NS
	Z	PM	PS	PS	Z	NS	NS	NM
	PS	PS	PS	Z	NS	NS	NM	NM
	PM	PS	Z	NS	NS	NM	NM	NL
	PL	Z	NS	NS	NM	NM	NL	NL

Fig. 6 – Tabla de reglas

5. Arquitecturas utilizadas

Como en todo FLC, se han estructurado las arquitecturas a partir de los procesos inherentes al control difuso, es decir: 1 - Fusificación, 2 - Inferencia Difusa y 3 - Defusificación.

Fusificación: con la información acusada por los sensores (datos "crisp"), se analiza el grado de pertenencia a cada una de las siete categorías para las dos variables de entrada (*ángulo* y *velocidad*). Este procedimiento devuelve en sendos registros, los valores mencionados y es común a todas las arquitecturas no obstante diferir según el caso la cantidad de cálculos efectuados.

Inferencia Difusa: una vez que se han fusificado las dos variables de entrada, se dispone en los registros de valores difusos que serán empleados para el proceso de inferencia difusa. En este caso, el mecanismo usado es la inferencia de Mamdani, aplicando funciones reticulares min-max: para toda función de pertenencia de salida se debe calcular la contribución que realiza cada valor difuso de entrada aplicando las reglas.

Para la solución del péndulo invertido aquí presentada, se ha elaborado una base consistente de 49 reglas que surgen de analizar todos los comportamientos para toda combinación posible de funciones de pertenencia de la variable ángulo y de la variable velocidad. La inferencia de cada regla particular se asegura aplicando la función mínimo a los valores correspondientes y la composición para cada función de pertenencia de salida se efectúa a través de la función máximo. En estas condiciones se obtienen los valores difusos para cada una de las siete categorías de fuerza que debe aplicarse para equilibrar al péndulo. Cabe aclarar que dichas funciones se han implementado como singleton.

Defusificación: resta, entonces, traducir dichos valores en una acción de control neta, es decir aplicar el proceso de defusificación a los valores antes mencionados. Con este propósito se emplea la técnica del centroide para lo cual se aproxima el centro de gravedad de cada función de pertenencia y se obtiene la media de dichos valores.

Para llevar a cabo esta tarea, se debe calcular el singleton de cada valor difuso de fuerza, lo cual consiste sencillamente de un producto. El cociente entre la sumatoria de estos siete valores y la sumatoria de los valores difusos devuelve el valor "crisp" de fuerza, o sea la acción de control sobre el carro del péndulo. Dicha fuerza se traduce en la práctica como un incremento (o decremento) en la velocidad del motor que acciona el desplazamiento del carro.

5.1. Arquitectura Pasiva

En la mencionada arquitectura, una vez obtenidos los datos "crisp", es decir los valores medidos de la inclinación del péndulo respecto de la vertical y su velocidad angular, se fusifican según dos métodos diferentes:

- 1 - Calculando en línea los valores de las funciones de pertenencia.
- 2 - Almacenando los valores de las funciones de pertenencia en memoria ROM (cálculo fuera de línea).

Para el primer enfoque se dispone de un banco de registros en los cuales se almacena los grados de pertenencia a cada una de las siete categorías de inclinación y a cada uno de los siete grados de velocidad angular. De aquí que se disponga de catorce valores difusos y tal como se detallara anteriormente, se les aplica el proceso de inferencia difusa, evaluando *totalmente* las 49 reglas.

Cuando se implementa el sistema de acuerdo a la segunda alternativa (memoria ROM), tanto las funciones de pertenencia de entrada como las de salida (singleton), se hallan calculadas fuera de línea y almacenadas en memoria lo cual redundo en un ahorro de tiempo a expensas de un incremento en la memoria requerida.

Respecto de la defusificación, cabe señalar que en el primer caso, los valores de los singleton se calculan en línea mientras que en el segundo caso se hallan almacenados en una memoria. En ambas arquitecturas se realiza el cociente antes descripto.

5.2. Arquitectura Activa

Fusificación: En esta alternativa se explota la posibilidad de calcular sólo las funciones de pertenencia que sean necesarias para la fusificación de una variable dada. Siempre es posible conocer cual es el grado de solapamiento entre funciones lo cual permite establecer cuantas serán no nulas a partir del momento en que se encontró la primera y de esta manera no se necesita calcular todo el dominio de la variable de entrada lo que produce un ahorro considerable en los tiempos de fusificación e inferencia difusa a expensas de una estructura más compleja para el cálculo.

Inferencia Difusa: El "banco de reglas" o "base de reglas" se denomina frecuentemente FAM (por *Fuzzy Associative Memory*) en la literatura [Chu96]. La asociación difusa del par (A , B) vincula la salida del conjunto difuso B (del valor de control) con la entrada del conjunto difuso A (de los valores de entrada), donde el par representa la asociación como antecedente – consecuente de una cláusula *IF-THEN*. El subconjunto de funciones de pertenencia activado, "disparará" a su vez a un subconjunto de reglas de la FAM, lo cual evita el tener que considerar a todas las reglas para la obtención de los valores difusos de salida. En [Aco01] se detalla la arquitectura activa y cómo se llevan a cabo las diferentes etapas involucradas las cuales se basan en determinar qué regla aplicar según los identificadores de las funciones de pertenencia. Así, se calcula la inferencia de dicha regla utilizando un banco de registros indexado por el consecuente de la regla activada. Este procedimiento se utiliza para las 4 reglas que potencialmente pueden activarse.

Defusificación: Una vez almacenados en los registros respectivos los valores difusos de salida se aplica el mismo proceso de defusificación descrito en la arquitectura pasiva. Aquí, al igual que en aquella, las funciones de pertenencia se pueden calcular en línea o fuera de línea almacenándolas en memoria ROM.

5.3 LUT (Lookup Table)

Esta arquitectura es la más sencilla y consiste de almacenar en una memoria el producto cartesiano de los dominios de ambas variables de entrada. En estas condiciones la fusificación, inferencia y defusificación se efectúan off-line y se almacenan en una memoria ROM de $181 * 61 \approx 11$ KB.

Conceptualmente, es la arquitectura más sencilla y veloz. La limitación recae en que el tamaño crece drásticamente al incrementar la cantidad de variables de entrada y la precisión de las mismas.

6. Conclusiones y trabajo futuro

El control difuso se efectuó sobre dos PC's diferentes: una con procesador Pentium corriendo a 200 MHz con 32 MB de memoria RAM y la otra con procesador Pentium III a 850 MHz con 256 MB de RAM, ambas ejecutando bajo Windows 98. La conexión con el dispositivo físico péndulo invertido se efectúa a través del puerto paralelo.

El carro se montó sobre dos rieles de acero y se controla por medio de un motor paso a paso, cuyo ángulo de giro por paso es de 15° (es decir 24 pasos por revolución). Sobre el carro se ha montado el péndulo (varilla plástica de 40 cm de longitud) que gira sobre un eje solidario a un encoder Hewlett Packard² encargado de medir el ángulo.

La velocidad del motor se controla por interrupciones del reloj de tiempo real de la computadora y el encoder genera interrupciones que permiten medir el ángulo (a una interrupción por grado).

El software de control difuso se ha estructurado según dos módulos principales:

- un módulo de manejo de hardware (driver) que consiste en el controlador del motor paso a paso y el medidor del ángulo que atiende la interrupción del encoder.
- software de control difuso (para cada uno de los cinco tipos de arquitecturas propuestas).

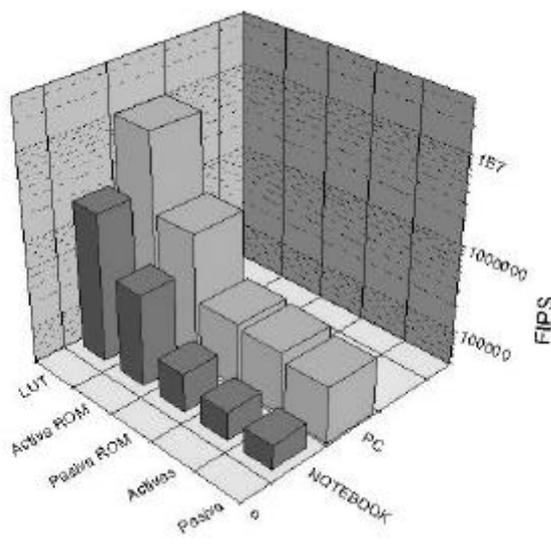


Fig. 7 – Fuzzy Inferences Per Second (FIPS) para el péndulo invertido

² - Modelo HEDS-5701-G00.

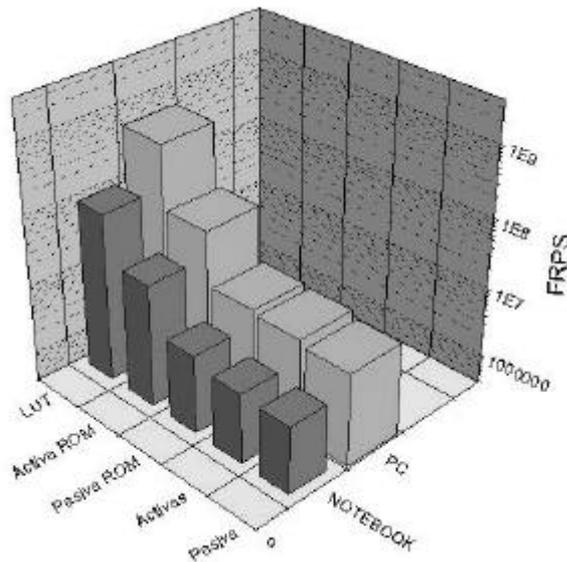


Fig. 8 – Fuzzy Rules Per Second (FRPS) para el péndulo invertido

Las gráficas (Fig. 7 y Fig. 8) se hallan expresadas en escala logarítmica, como se puede observar la implementación con Lookup Tables posee un factor de 100 veces más velocidad que las anteriores arquitecturas evaluadas. Se presentan los valores obtenidos para los procesadores antes mencionados, refiriéndose la gráfica rotulada "NOTEBOOK" al ordenador con procesador Pentium 200, mientras que la gráfica rotulada "PC" se refiere a aquel con procesador Pentium III.

De los resultados se advierte que las implementaciones con la arquitectura activa permiten obtener tiempos de respuesta elevados respecto de la arquitectura pasiva. En particular cuando se utiliza un enfoque orientado al cálculo es más significativa la diferencia entre ambas arquitecturas (es decir cuando no se utiliza un soporte de funciones de pertenencia en tablas). En este caso se advierte que, el hecho de calcular sólo las funciones que se activan, aumenta el throughput en aproximadamente un factor 5.



Fig. 9 - Sistema físico que implementa al péndulo invertido

Actualmente se está realizando un análisis similar al aquí presentado sobre plataformas Linux y Real Time Linux. Se encuentra en fase de diseño un controlador hardware a medida sobre FPGA.

Bibliografía

- [Aco00] “*Motores de Inferencia para Controladores Difusos: análisis de materializaciones hardware*”. Congreso Internacional en Ing. Informática (ICIE). UBA, Bs. As. 26-28 abril 2000. Acosta N. & Curti H.
- [Aco01] “*Controladores Difusos de Altas Prestaciones: sistema para la generación automática*”, Acosta N. Workshop de Investigadores en Ciencias de la Computación (WICC), Potrero de Funes, San Luis. 22-24 mayo 2001.
- [Aco98] “*Implementación de controladores difusos sobre FPGAs*”, N. Acosta, J-P Deschamps y E. Todorovich. XXVI Congreso Argentino de Control Automático, organizado por la Asociación Argentina de Control Automático, Buenos Aires, Argentina. Ago'98.
- [Cha01] “*Design of robust fuzzy-model-based controller with sliding mode control for SISO nonlinear systems*”, W. Chang, JB Park, YH Joo & G. Chen. Fuzzy Sets and Systems, (2001). A publicarse próximamente.
- [Che99] “*Fuzzy predictive control of uncertain chaotic systems using time series*”, L. Chen & G. Chen. Int. Journal of Bifurcation and chaos, Vol. 9, Nro. 4. (1999). Pp: 757-767.
- [Chi92] “*Optimization of fuzzy logic inference architecture*”, T. C. Chiueh. Computer, pp: 67-71, May 1992.
- [Chu96] “*On Fuzzy Associative Memory with Multiple-rule Storage Capacity*”, Chung F. and Lee T.. IEEE Trans. on Fuzzy Systems, vol. 4, Nro. 3, August, pp: 375-384. (1996)
- [Coo94] “*Genetic design of fuzzy controllers: the cart and joined-pole problem*”, M. Cooper & J. Vidal. Third IEEE Int. Conf. on Fuzzy Systems, Orlando, Florida. June 1994.
- [Des97] “*Metodología de síntesis de controladores difusos*”, Deschamps J-P & Bioul Géry. III Workshop Iberchip, CINVSTAV, Mexico D.F., 1997.
- [Des98] “*Fuzzy controller synthesis method*”, Jean Pierre Deschamps, pp:231-255. Artículo publicado en: “*Fuzzy hardware: Architecture & applications*”, A. Kandel y G. Langholz. Kluwer Academic Publishers, 1998.
- [Jag95] “*Fuzzy logic in control*”, René Jager. PhD Thesis. Technische Universiteit Delft, 1995.
- [Jan95] “*Neuro-fuzzy modeling & control*”, J. Jang & C. Sun. Proc. of the IEEE, Vol. 83, Nro. 3, pp: 378-406. March'95.
- [Kos94] “*Function approximation with additive fuzzy systems*”, B. Kosko y J. A. Dickerson, capítulo 12 en “*Theoretical aspects of fuzzy control*” de H. Nguyen, M. Sugeno y R. Yager Eds., NY, Willey, 1994.
- [Lee01] “*Robust fuzzy control of nonlinear systems with parametric uncertainties*”, H Lee, B Park & G Chen. IEEE Trans. on Fuzzy Systems, Vol. 9, Nro. 2, April 2001. Pp: 369-378.
- [Sch00] “*Qualitative modelling using dynamic fuzzy systems*”, K. Schmid & V. Krebs. 3rd. Int. Symp. on Mathematical Modelling, Vienna, Austria, Feb 2-4 (2000).
- [Wan96] “*Stable adaptive fuzzy controllers with application to inverted pendulum tracking*”, Li-Xin Wang. IEEE Trans. on Systems, Man & Cybernetics, part B, Vol. 26, Nro. 5. Oct'96.