

EVALUACIÓN DE LA APLICACIÓN DE TÉCNICAS DE LÓGICA DIFUSA EN EL CONTROL DE MÁQUINAS ELÉCTRICAS

Jaime F. Fonseca

João L. Afonso

Júlio S. Martins

Carlos A. Couto

Departamento de Electrónica Industrial

Universidade do Minho

Largo do Paço

4709 – Braga Codex

Portugal

RESUMEN

Este artículo presenta un trabajo desarrollado para una evaluación de la aplicación de técnicas de lógica difusa (fuzzy) al control de máquinas eléctricas, y más específicamente a un motor de inducción trifásico. Se realiza una introducción del Matlab/Simulink, utilizado para desarrollar y simular todo el sistema basado en la técnica referida. Asimismo se presenta la herramienta informática fuzzyTECH, empleada para desarrollar, depurar y posteriormente generar el código correspondiente al sistema de control. En este trabajo se compara la ejecución de un regulador fuzzy con un regulador PI y se demuestra que, mediante una elección adecuada de las reglas y de las funciones pertenencia, su comportamiento dinámico es mejor, principalmente, en términos de la insensibilidad a la variación de parámetros. Los resultados obtenidos permite inferir que las técnicas de lógica difusa es posible en este tipo de aplicaciones siendo una alternativa interesante a las técnicas de control clásico.

Palabras clave: Matlab/Simulink, fuzzyTECH, regulador Fuzzy, regulador PI, controlador de deslizamiento.

EVALUATION OF FUZZY LOGIC TECHNIQUES APPLIED TO THE CONTROL OF ELECTRICAL MACHINES

ABSTRACT

This paper reports work that is being developed in the evaluation of fuzzy logic techniques applied to the control of electrical machines, using the induction motor as an example. An overview of a CACSD (Computer Aided Control System Design) tools is also presented: Matlab/Simulink, used to design and simulate a control system based on the referred techniques; *fuzzyTECH*, used to design, tune and subsequently generate the code which implements fuzzy controller. The evaluation of the fuzzy logic controller behaviour is made through the comparison with a traditional PI controller and it will be shown that, with a proper selection of membership functions, its dynamic response is better, namely in terms of robustness. From the results obtained, it can be concluded that the use of fuzzy logic control for this kind of applications is not only viable, but also an interesting alternative to more conventional techniques.

Keywords: Matlab/Simulink, fuzzyTECH, Fuzzy controller, PI controller, slip controller.

INTRODUCCION

En los últimos años la lógica difusa (fuzzy) ha aparecido como una técnica alternativa en el control de procesos industriales complejos, y en las más diversas instalaciones electrónicas y de electrodomésticos. La lógica fuzzy es un superconjunto de la lógica clásica convencional que fue extendida para abarcar el concepto de "parcialmente verdad" (valores entre lo "absolutamente cierto" y lo "absolutamente erróneo"). Fue introducida en 1960 por el Dr. Lofti Zadeh de la UC/Berkeley, para modelar el conocimiento subjetivo, que se basa en informes lingüísticos normalmente imposibles de cuantificar usando los métodos matemáticos comunes. La utilización de la lógica difusa permite al diseñador procesar eficazmente los conocimientos objetivos y subjetivos (datos numéricos y conocimiento expreso de expresiones lingüísticas), aplicados a los problemas de control en bucle cerrado, disminuyendo así el tiempo de desarrollo del proyecto y los costes (Mendel, 1995).

Hasta el momento, en automatización industrial la lógica difusa ha sido utilizada principalmente en procesos relativamente lentos (Guillemin, 1996). Este artículo presenta el trabajo desarrollado en el intento de aplicar estas técnicas a procesos rápidos, en particular para el control de máquinas eléctricas, y más concretamente para un motor de inducción trifásico, mediante la utilización de circuitos específicos, es decir un regulador fuzzy. Se ha dado especial énfasis a las herramientas informáticas de desarrollo utilizadas como Matlab y fuzzyTECH.

SIMULACION Y DESARROLLO

La simulación de los sistemas físicos es de gran importancia en los campos teóricos y/o aplicados. La simulación facilita a los científicos e ingenieros el análisis rápido del comportamiento de los sistemas físicos, definidos por el correspondiente esquema de bloques o del modelo matemático, permitiéndoles así una disminución de los tiempos de desarrollo con una consecuente reducción de los costes.

Las herramientas de simulación y desarrollo tienen un papel muy importante cuando se diseñan sistemas de control, especialmente en el caso de los reguladores fuzzy, pues éstos son especificados según la experiencia subjetiva proporcionada por expertos y por consiguiente la depuración es enteramente dependiente de los medios de simulación.

Matlab es un entorno que combina el análisis numérico, la visualización gráfica y lenguaje de alto nivel. Es un entorno natural para análisis, desarrollo de algoritmos y de aplicaciones. Matlab dispone de un conjunto de librerías que presenta

funciones para solucionar problemas particulares, denominadas "Toolboxes".

En este caso fueron utilizadas las siguientes "toolboxes" del Matlab: "Nonlinear Control Design Toolbox" para armonizar el regulador PI con el mecanismo de reset windup e "Identification Toolbox" para la obtención del modelo conjunto inversor-motor de inducción trifásico.

Simulink está construido sobre Matlab, proporcionando un entorno interactivo para modelación, análisis y simulación de una gran variedad de sistemas dinámicos.

La herramienta fuzzyTECH MCU-96 edition, utilizada para desarrollar el regulador fuzzy, cubre todas las fases del proyecto de un regulador de este tipo. Esta herramienta permite definir el proyecto, las variables lingüísticas y las reglas, así como la realización de una simulación interactiva. También facilita la generación de código ANSI C, Kernighan and Richie C y ensamblador, en este caso para el microcontrolador INTEL de la familia 96. Versiones de fuzzyTECH para otros microcontroladores también se encuentran disponibles.

CONTROL DEL MOTOR DE INDUCCION

La figura 1 presenta el esquema convencional de un controlador de deslizamiento para un motor de inducción (Martins,1993; Couto y Martins,1994), utilizado en variadores de velocidad sin gran exigencia de prestaciones. Tradicionalmente, el error de velocidad (ω_e) es la entrada del regulador PI, que establece la frecuencia de deslizamiento del motor (ω_r).

La frecuencia estática resulta de la adición de la frecuencia de deslizamiento a la velocidad del rotor (ω). La tensión del estator (U_s) es establecida de acuerdo con una ley predefinida (U_s/ω_s aproximadamente constante), de forma que el flujo del motor sea mantenido en su valor nominal. La frecuencia de deslizamiento es limitada, estableciendo (indirectamente) un límite para el par máximo disponible en el motor y para la corriente del estator.

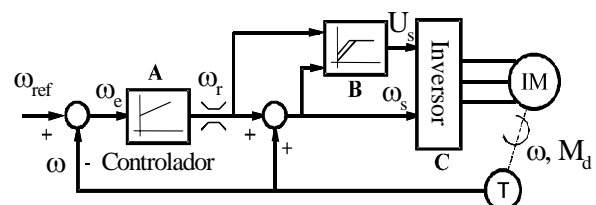


Fig. 1: Esquema convencional de un controlador de deslizamiento

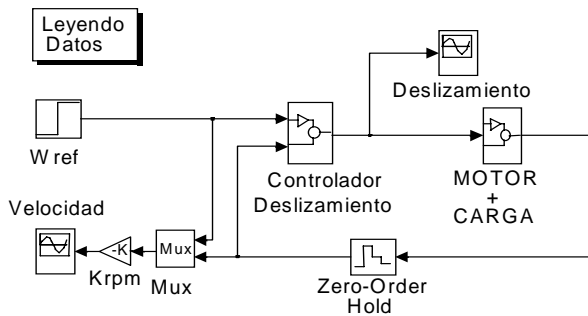


Fig. 2: Esquema de bloques usado en el Simulink

La idea presente fue simulada en el entorno Matlab, utilizando técnicas fuzzy para diseñar el regulador. Los resultados obtenidos son comparados con los de un regulador clásico (PI con mecanismo de reset windup) (Fonseca et al., 1996).

La figura 2 presenta el esquema de bloques del sistema usado en el entorno Simulink (Cavallero et al., 1996). Esta herramienta tiene una librería con diferentes bloques, diversos algoritmos de integración y permite al usuario una fácil selección de los parámetros de simulación.

El bloque "Controlador de Deslizamiento" contiene todo el cálculo numérico y el regulador fuzzy necesarios para la realización del controlador de deslizamiento, según se representa en la figura 3. El bloque "Motor + Carga" contiene el modelo del motor de inducción y de la carga determinados mediante la "Identification Toolbox".

El regulador fuzzy fue realizado mediante fuzzyTECH, que genera el código M (específico de Matlab), que permite tratarlo en el Simulink como si fuese una función M.

Fue utilizado fuzzyTECH en lugar de la toolbox de lógica fuzzy Matlab por los siguientes motivos:

- *FuzzyTECH* permite generar código optimizado para el microcontrolador Intel 80C196KC ya que se pretende que la simulación sea lo más cercana posible a la situación real;
- la toolbox de fuzzy proporcionada por Matlab no tiene incorporado el método de lógica difusa inversa que posibilitaría la obtención de un código más optimizado (CoM - *Center of Maximum*).

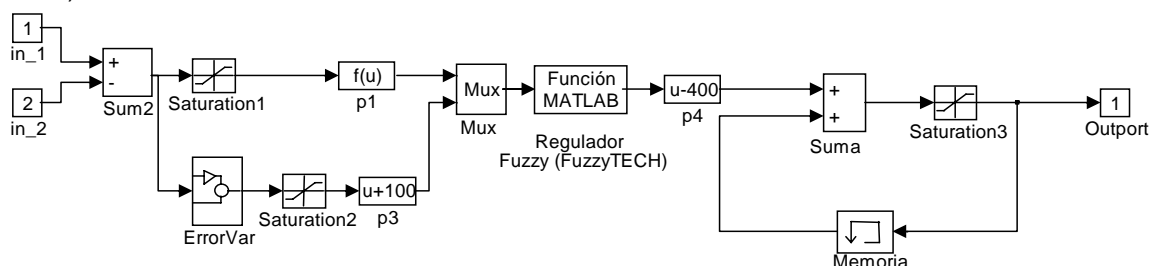


Fig. 3: Esquema de bloques del controlador de deslizamiento

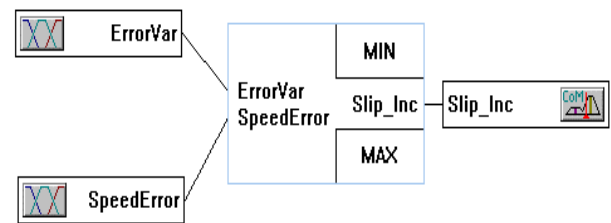


Fig. 4: Estructura del controlador de deslizamiento para el motor de inducción

El esquema de bloques utilizado en la simulación del regulador PI es idéntico al representado en la figura 2, solamente el bloque del controlador de deslizamiento es sustituido por un bloque que ejecuta al regulador PI. Cabe destacar que el PI fue depurado utilizando otra toolbox de Matlab denominada Nonlinear Control Design.

DESARROLLO DEL REGULADOR FUZZY

Cómo ya se mencionó anteriormente, la herramienta fuzzyTECH MCU-96 Edition dispone de todas las etapas necesarias para el desarrollo de un regulador fuzzy (Von Altrock, 1995; Inform Software Corporation, 1996).

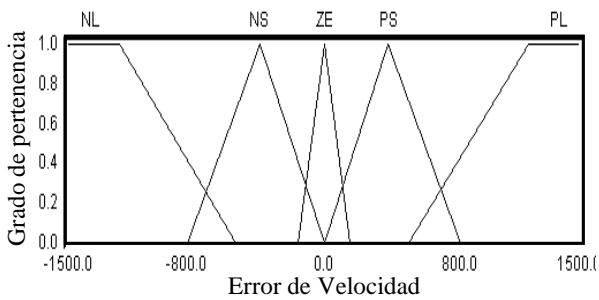
Definición del proyecto

El primer paso cuando se usa fuzzyTECH MCU-96 Edition es definir la estructura del regulador por medio de la ventana Project Editor.

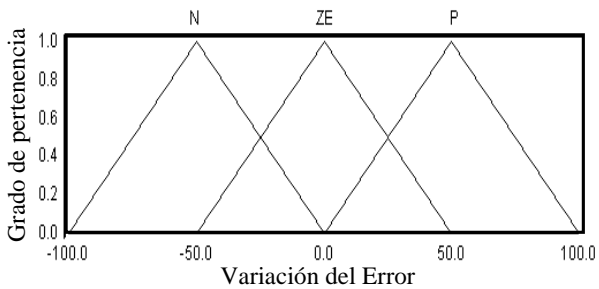
La figura 4 presenta la estructura del regulador y permite al diseñador el acceso directo a la definición de las variables lingüísticas y de las reglas.

Definición de las variables lingüísticas

La etapa siguiente consiste en la definición de las variables lingüísticas. La interfaz gráfica de fuzzyTECH facilita al diseñador la creación de las variables lingüísticas más comunes y las funciones pertenencia para la aplicación. El controlador de deslizamiento, representado en la figura 4, tiene dos entradas, Error de Velocidad (*SpeedError*) y la Variación del Error (*ErrorVar*), y una salida, que es Incremento del Deslizamiento (*Slip_Inc*).



(a)



(b)

Fig. 5: Funciones pertenencia para: (a) Error de Velocidad (SpeedError); (b) Variación del Error (ErrorVar)

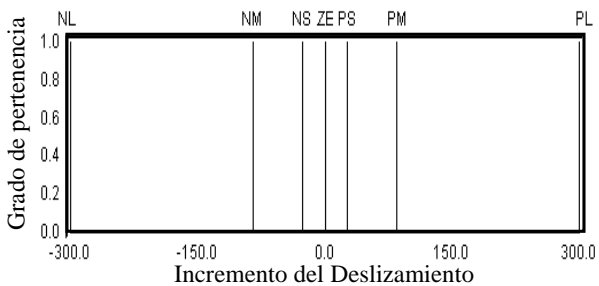


Fig. 6: Funciones pertenencia para el Incremento del Deslizamiento (SlipInc)

El Error de Velocidad (figura 5a) es descrito por 5 funciones pertenencia triangulares: negativo grande (NL), negativo pequeño (NS), cero (ZE), positivo pequeño (PS) y positivo grande (PL). La variación del Error (figura 5b) es descrita por 3 funciones pertenencia triangulares: negativo (N), cero (ZE) y positivo (P). El Incremento del Deslizamiento (figura 6) es descrito por 7 funciones pertenencia triangulares: negativo grande (NL), negativo medio (NM), negativo pequeño (NS), cero (ZE), positivo pequeño (PS), positivo medio (PM) y positivo grande (PL).

La función de pertenencia determina el grado de pertenencia de un valor del universo de discurso de una variable a un determinado conjunto borroso. Este grado varía entre 0 y 1. Como función de pertenencia se suelen emplear por su sencillez funciones triangulares centradas en un determinado valor y con una determinada anchura. Las funciones pertenencia de entrada presente en las figuras 5 y 6 son definidas teniendo en cuenta

la velocidad, la aceleración del motor y la resolución del sistema.

Durante la definición de las variables lingüísticas, fuzzyTECH permite al usuario definir dos representaciones para las variables: shell values y code values. Los primeros son los valores del mundo real que las variables representan, siendo solamente usadas para presentar los valores actuales con fuzzyTECH. Los segundos son los valores internos de 16 bits que el microcontrolador usa para calcular los resultados, y el margen está entre 0 y 65535. Si la escala para los code values y para los shell values fuese la misma, sería más fácil percibir el compartamiento del control por comparación directa de las entradas/salidas fuzzy real con las variables lingüísticas.

Definición de las reglas

La figura 7 representa al Editor de Reglas con las reglas definidas para el controlador de deslizamiento. La definición de las reglas permite abarcar mejor el compartamiento del sistema. Fueron definidas reglas para: mantener el error de velocidad (SpeedError) a la vuelta de cero, evitar el sobrepulso (overshoot) en la velocidad del motor y las que proporcionan una rápida respuesta cuando el error de velocidad es grande. La relación entre los antecedentes de la regla es un Y lógico. La construcción de la tabla se realiza en base a la experiencia de un operador del proceso.

Spreadsheet Rule Editor - RB1				
	IF		THEN	
	SpeedErr	SpeedErrVar	DoS	Slip_Inc
1	NL		1.00	NL
2	NS	N	1.00	NM
3	ZE	N	1.00	NS
4	PS	N	1.00	NM
5	NS	ZE	1.00	NS
6	ZE	ZE	1.00	ZE
7	PS	ZE	1.00	PS
8	NS	P	1.00	PM
9	ZE	P	1.00	PS
10	PS	P	1.00	PM
11	PL		1.00	PL
12				

Fig. 7: Editor de reglas

Optimización del comportamiento del sistema

La herramienta fuzzyTECH proporciona mecanismos fuera de línea (off-line) para depuración (debug), prueba y optimización de las reglas y funciones de pertenencia. El modo de depuración interactiva ofrece una verificación gráfica de todas las etapas del proyecto. El modo batch permi-

te crear un archivo de salida asociado a un archivo de entradas que contiene una muestra de los valores que éstas pueden tener. Este modo permite analizar y ensayar la ejecución del proyecto.

Asimismo fuzzyTECH presenta un mecanismo de depuración en línea (debug on-line) mediante comunicación RS-232.

Usando estas características, el diseñador puede examinar si todas las reglas establecidas son necesarias, si las reglas importantes no han sido omitidas y si las variaciones de las variables de salida son consistentes con el sistema.

El código para el microcontrolador Intel 80C196KC es generado automáticamente por fuzzyTECH.

RESULTADOS

A continuación se presentan y comparan los resultados de la simulación de un regulador fuzzy y de un regulador clásico (PI con el mecanismo de reset windup), sintonizado con la Nonlinear System Design Toolbox, en respuesta a un escalón de carga de 0% para 100% del par nominal (figura 8).

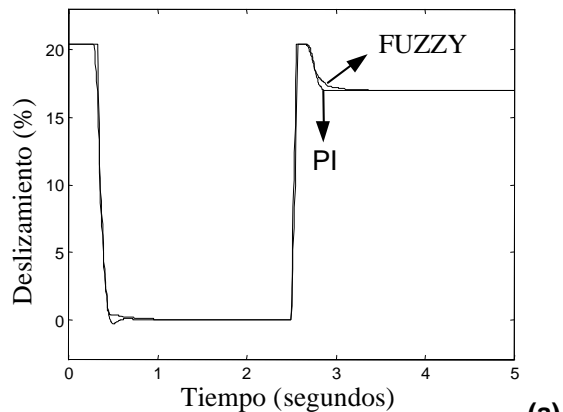
La figura 9 presenta la velocidad del motor y el deslizamiento durante el arranque con 10% del par nominal y $J = 2.07 \times 10^{-4} \text{ kg.m}^2$ cuando es usado respectivamente los reguladores fuzzy y PI. En este caso la respuesta de ambos reguladores es semejante.

Todas las simulaciones fueron realizadas con un periodo de muestras igual a 5 ms.

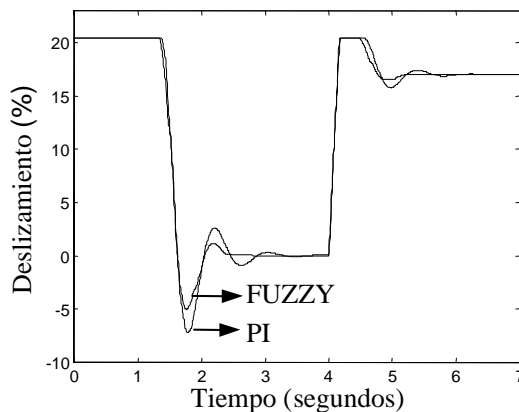
CONCLUSIONES

Los resultados de la simulación, utilizando las herramientas Matlab/Simulink y fuzzyTECH, demuestran que el comportamiento dinámico de un regulador con lógica fuzzy es mejor que el de un regulador PI clásico, principalmente en lo que se refiere a la insensibilidad frente a alteraciones de los parámetros del modelo.

Otra ventaja del regulador fuzzy, que cabe señalar, es la facilidad de diseño ya que la estrategia de control se realiza en base al conocimiento de un operador del proceso. No obstante, presenta el inconveniente de que no existe una metodología de ajuste del regulador, realizándose el ajuste mediante un procedimiento de prueba y error.

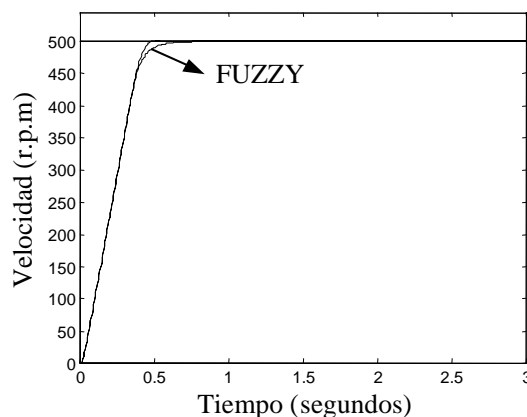


(a)

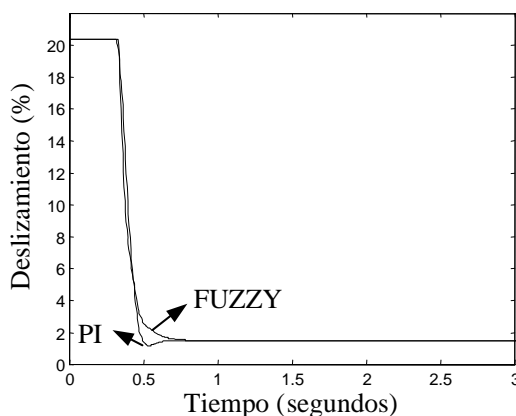


(b)

Fig. 8: Respuesta a un escalón del par: (a) $J = 2.07 \times 10^{-2} \text{ kg.m}^2$; (b) $J = 8.28 \times 10^{-2} \text{ Kg.m}^2$



(a)



(b)

Fig. 9: Arranque del motor: (a) velocidad; (b) deslizamiento

AGRADECIMIENTOS

Los autores expresan su agradecimiento a Heriberto Hernandez Martinez y al profesor Perfecto Mariqo Espiqueira, de lo Departamento de Tecnologia Electronica de la Universidad de Vigo, su colaboración en la revisión de la versión en Español de lo artículo.

REFERENCIAS

Cavallo A., Setola R., Vasca F., Using Matlab, Simulink and Control System Toolbox: a Pratical Approach, Prentice-Hall (1996).

Couto C. and Martins J. S., Control of a Voltage Source Inverter Fed Induction Motor with On-Line Efficiency Optimisation, IEEE ICIT'94, Guangzhou, China, 528-532, (1994).

Fonseca J., Afonso J. L., Martins J. S., Couto C., Evaluation Of Neural Networks and Fuzzy Logic Techniques Applied to the Control Of Electrical Machines, Proceedings of the Mechatronics'96 (July 1996).

Guillemin P., Fuzzy Logic Applied to Motor Control, IEEE Transaction on Industry Applications, Vol. 32, No. 1, pp 51-56 (January/February 1996).

Inform Software Corporation, *fuzzyTECH* Reference Manual, Inform GmbH (1996).

Martins J. S., Controlo de Velocidade do Motor de Indução Trifásico, PhD Thesis, Minho University, Portugal (1993).

Mendel J. M., Fuzzy Logic Systems for Engineering: A Tutorial, Proceedings of the IEEE, Vol. 83, No. 3, pp 345-377 (1995).

Von Altrock C., Fuzzy Logic & NeuroFuzzy Applications Explained, in: Prentice-Hall (1995).