ISSN: 1692-7257 - Volumen 2 – Número 20 - 2012

Revista Colombiana de
Tecnologías de Avanzada

Recibido: 01 de junio de 2012
Aceptado: 30 de junio de 2012

APPLICATION OF NEURAL NETWORKS TO AC MOTORS SPEED CONTROL

APLICACIÓN DE REDES NEURONALES AL CONTROL DE VELOCIDAD EN MOTORES DE CORRIENTE ALTERNA

Ing. Carolina Martínez Quintero, MSc. Jorge Luis Díaz Rodríguez
PhD. Aldo Pardo García

Universidad de Pamplona, Grupo de Investigaciones en Automatización y Control.
Ciudadela Universitaria. Pamplona, Norte de Santander, Colombia.
Tel.: +57-7-568 5303 Ext. 164, Fax: +57-7-568 5303 Ext. 156.
E-mail: {caromaqui, jdiazcu}@gmail.com, apardo13@hotmail.com

Abstract: This paper establish a comparison of the application of traditional methods of AC motors speed control, and controllers based on artificial neural network techniques, including PI and PID control, PID Cloned, Predictive and Narma-L2 neural controllers. We present the results of controlling the speed of the induction motor using the Matlab professional software.

Keywords: Induction Motor, electric drives, speed control, PID, neural networks.

Resumen: En este artículo se establece una comparación de la aplicación de métodos tradicionales en el control de velocidad de motores de corriente alterna y controladores basados en técnicas de redes neuronales artificiales, que incluyen el control PI y PID, y los controladores neuronales PID Clonado, Predictivo y Narma L2. Se presentan los resultados de controlar la velocidad del motor de inducción empleando la herramienta de simulación profesional.

Palabras clave: Motor de inducción, accionamiento eléctrico, control de velocidad, PID, redes neuronales.

1. INTRODUCCIÓN

Con el uso de Inteligencia Artificial en los campos de electrónica de potencia y en los accionamientos eléctricos se han incrementado considerablemente. Estas técnicas modernas basadas principalmente en sistemas expertos, lógica difusa y redes neuronales son utilizadas para el control y facilitan la toma de decisiones desde un análisis lógico basado en conocimientos previos o datos que describen el funcionamiento del sistema.

En este estudio se presenta la aplicación de métodos tradicionales en el control de velocidad de motores eléctricos utilizando, controladores de tipo PID, comparados con una de las técnicas de inteligencia artificial: las redes neuronales.

Los datos necesarios para entrenar las redes neuronales artificiales se obtienen, en primer lugar, de los resultados de las simulaciones de los controladores PI y PID, y en segundo lugar, a partir de los métodos experimentales (prueba y error).

Luego de realizadas las simulaciones de los controladores convencionales (PI, PID) y los

neuronales PID clonado, predictivo y narma L2, se realiza una comparación entre los resultados de utilizar control clásico versus los controladores basados en técnicas de redes neuronales.

2. CONTROL CLÁSICO

2.1 Sistema en lazo abierto

El sistema a lazo abierto se excita mediante una entrada escalón unitario y de esta forma se puede analizar la respuesta transitoria y estacionaria. En la figura 1 se muestra el sistema en lazo abierto, está compuesto por el motor de inducción, el convertidor, y las diferentes salidas del sistema [Pardo y Díaz, 2004], en nuestro caso para el análisis frente a un escalón unitario sólo se utiliza la respuesta de velocidad.

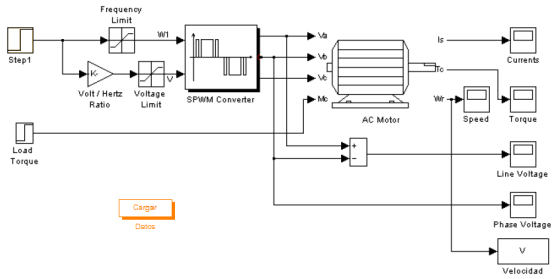


Fig. 1. Diagrama de simulación en lazo abierto del motor de inducción.

El sistema presenta estabilidad sin sobreregulación a una entrada escalón unitario, como se puede observar en la figura 2, motivo por el cual podemos implementar el método primer método de Ziegler Nichols (curva de reacción) de donde se pueden extraer el tiempo de retardo L y la constante de tiempo T. El tiempo de retardo y la constante de tiempo se determinan dibujando una recta tangente en el punto de inflexión de la curva.

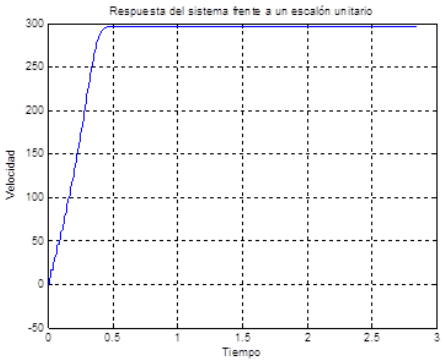


Fig. 2. Respuesta del sistema al escalón unitario.

De donde de obtiene, T = 0.23 seg y L = 0.06 seg.

Los valores de K_p, T_i, en el caso de un regulador PI, se utilizaron los de la tabla 1 según la regla de sintonía Ziegler y Nichols.

Tabla 1: Ajuste Ziegler y Nichols.

Tipo de Controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9\frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2\frac{T}{L}$	$2L$	$0.5L$

Con los valores de la constante de tiempo (K_p) y tiempo de retardo (T_i) para el PI se obtiene:

$$K_p = 0.9\frac{T}{L} = 3.45; \quad T_i = \frac{L}{3} = 0.2; \quad T_d = 0$$

$$G_{PI}(s) = K_p\left(1+\frac{1}{T_i\, s}+T_d\, s\right) = 3.45\left(1+\frac{1}{0.2\, s}\right)$$

2.1 Controlador PI

Se adiciona un bloque de control proporcional PI, y se cierra el lazo de control haciendo uso de la realimentación del sistema, se aplica una referencia de 375 [rad/seg], en rpm serían aproximadamente 3580 rpm, para los datos del motor utilizado.

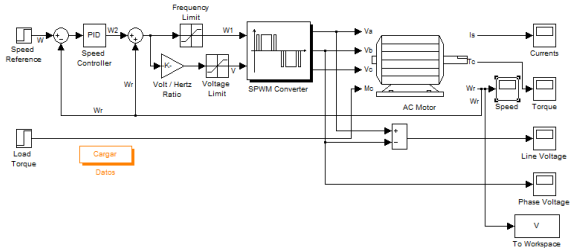


Fig. 3. Diagrama de simulación en lazo cerrado con controlador PI.

En la figura 3 encontramos el diagrama de simulación en lazo cerrado el cual está compuesto por el controlador PI, el motor que internamente ya hemos tratado las ecuaciones en capítulos anteriores y la estrategia PWM. En este diagrama se tomó la salida de velocidad para hallar el error, y así realimentar al controlador PI y posteriormente los demás controladores.

La figura 4 muestra la respuesta de velocidad utilizando el controlador PI. Al estabilizarse a la señal controlada de velocidad se aplicó una perturbación, donde se observa la estabilización y su capacidad limitada de rechazo al disturbio frente a la perturbación, sin poder volver a recuperarse.

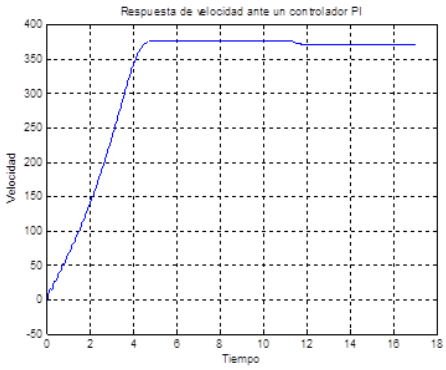


Fig. 4. Respuesta de velocidad con controlador PI.

Realizando un acercamiento de la figura 4 en ambos ejes se observa que la velocidad final a la que llega este controlador ante la perturbación queda en un valor aproximado a 370 rad/seg, teniendo en cuenta que el valor de referencia es 375 rad/seg, se evidencia un error en estado estable de 2 rad/seg.

2.2 Controlador PID

El modelo en lazo abierto desarrollado en la fig. 1 es utilizado para varios métodos de ajuste de controlador. Para un controlador PID y utilizando la regla de sintonía Ziegler-Nichols. Se determinan los valores de los parámetros del controlador PID, según la tabla 1 de la misma forma que el procedimiento anterior, realizando los cálculos para obtener las variables K_p, T_i, y T_d.

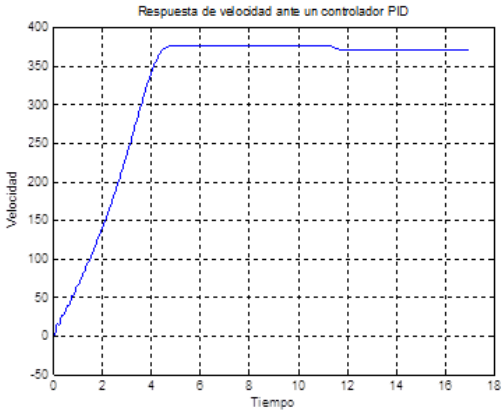


Fig. 5. Respuesta de velocidad - controlador PID.

Ajustados los parámetros del controlador PID, y realizada la simulación como se muestra en la figura 5, se obtiene la respuesta de velocidad la cual es muy similar a la salida del controlador PI (ver fig. 2). El sistema sigue siendo expuesto a una perturbación a los 11.5 segundos.

3. CONTROLADORES NEURONALES

3.1 Controlador neuronal PID clonado

Este modelo presenta el uso de redes neuronales en sustitución del bloque PID. Se diseña una RNA de tal forma que aproxime la respuesta del regulador, para esto se emplea *Simulink* de Matlab y su caja de herramientas *Neural Network Toolbox* (*NNT*).

Se requieren datos representativos del entorno en el cual funcionara (conjunto de entrenamiento), para realizar la implementación se deben obtener dichos datos. Esto se logra mediante el bloque "*To Workspace*" (*simout*) que se encuentra en la librería básica *Simulink/Sinks* y se usa para almacenar muestras de señales de Simulink generadas durante una simulación. Las muestras se asignan a una variable de Matlab, que son tomadas para el respectivo entrenamiento de la red, se toman 102066 muestras para lograr un conjunto representativo de todas las condiciones de operación presentes en la simulación. Para simular la red neuronal tomamos como data la entrada y su salida del controlador PID. La red es creada mediante el comando *newff* para creación de redes *backpropagation*, con el siguiente código:

red = *newff*(*minmax*(p),[s1 s2 s3], ...
{'*purelin*','*tansig*','*purelin*'});

Se toma s1=4 como el número de neuronas en la capa de entrada, s2=9 para la capa oculta y s3=1 en la capa de salida, estos valores son seleccionados aleatoriamente. Cabe resaltar que no existe una técnica para determinar el número de capas ocultas, ni el número de neuronas que debe contener cada una de ellas según el caso a desarrollar, es seleccionado por la experiencia del diseñador ó por ensaño y error. Se utiliza como función de transferencia *purelin* (lineal) y *tansig* (Tangente Sigmoidal Hiperbólica) como parámetros que determinan el entrenamiento, con lo cual se logró un rendimiento óptimo, basado en su respuesta.

Se genera la red y se realiza el respectivo remplazo del controlador convencional PID por el neuronal, lo que se puede observar en la figura 6.

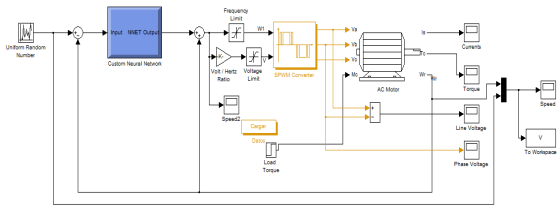


Fig. 6. Control neuronal PID clonado.

Como respuesta al sistema se tiene la figura 7.

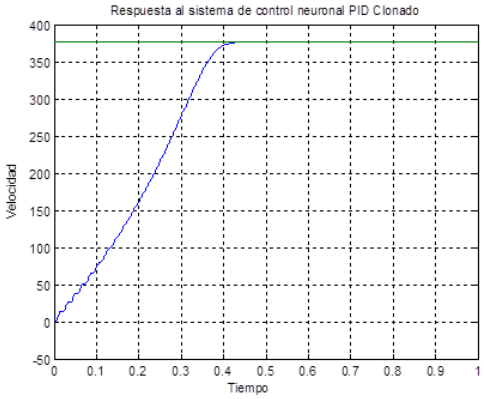


Fig. 7. Respuesta del sistema al control neuronal.

Una vez realizada la simulación en lazo cerrado del sistema de control del motor de inducción, y teniendo como estrategia de control el PID clonado se puede observar que el sistema, no presenta sobreregulación, el tiempo de respuesta transitoria es corto y no presenta error en estado estacionario.

3.2 Controlador neuronal predictivo

El modelo del controlador neuronal predictivo que se implementa en la *Toolbox* de Matlab utiliza una red neuronal del modelo de la planta no lineal para predecir el rendimiento de la planta en el futuro. El controlador calcula entonces la entrada de control que optimizará el rendimiento de la planta en un horizonte de futuro de tiempo especificado.

Como primer paso se realiza la identificación del sistema, es decir, se determina el modelo neuronal de la planta. Seguidamente, el controlador utiliza el modelo de la planta para predecir el rendimiento futuro.

Este bloque controlador predictivo de la *Neural Network Toolbox* de Matlab, establece que la señal de control se conecte a la entrada del modelo de la planta. La salida del modelo de la planta se conecta a la salida de la planta, y la señal de referencia se conecta a la referencia (ver figura 8).

Se establecen los parámetros de diseño en el bloque controlador predictivo NN de Matlab. En este se define la planta, se debe desarrollar primero el modelo neuronal de la planta antes de utilizar el controlador. El modelo de la planta predice futuros resultados de la planta. El algoritmo de optimización utiliza estas predicciones para determinar las entradas de control que optimicen el rendimiento futuro.

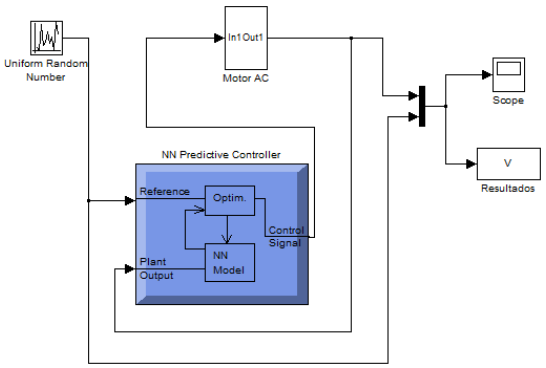


Fig. 8. Simulación del modelo del control predictivo en simulink de Matlab.

Para el modelo de la planta se definen 13 neuronas en la primera capa, intervalo de muestreo 0.2 segundos, 2 líneas de retardo en la entrada y salida de la planta. Se establecen como datos de entrenamiento a la red, número de muestras 10000, valor máximo y mínimo de entrada a la planta 400 y 150 respectivamente, se puede también seleccionar un rango en los datos de salida a ser utilizado en el entrenamiento.

Seguidamente se procede a generar los datos de entrenamiento, el programa los genera mediante la aplicación de una serie de entradas escalón al azar en el modelo de planta de simulink. Después de que el modelo de la planta ha sido formado, se acepta para cargar la red en el modelo de *Simulink*.

Cabe destacar que los parámetros establecidos se hacen de manera experimental, por ensayo y error, gracias a la estructura que utiliza Matlab brindando el acceso a todas las propiedades de la red, independientemente del tipo, permitiendo así su configuración según la necesidad.

Se utilizó la función *trainrp* (algoritmo *backpropagation resileint*) para el entrenamiento de la planta, utilizando los datos existentes para capacitar a la red. Después de que la formación es completada, se inicia la simulación del sistema, donde la salida de la planta y la señal de referencia se muestran en la figura 9.

La respuesta en lazo cerrado obtenido en el sistema con la estrategia de control neuronal predictivo presenta cambios bruscos y mayor tiempo en la respuesta transitoria, cabe resaltar que en la respuesta estacionaria el sistema no alcanza el valor de referencia o valor deseado presentando error en estado estacionario.

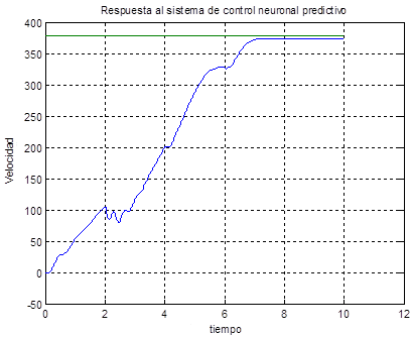


Fig. 9. Salida de la planta y señal de referencia frente al controlador predictivo.

3.3 Controlador Narma-L2 (Linealización por Realimentación)

Utilizando nuevamente la caja de herramientas de redes neuronales y *Simulink* de Matlab se crea el modelo del control Narma L2 que se puede observar en la figura 10.

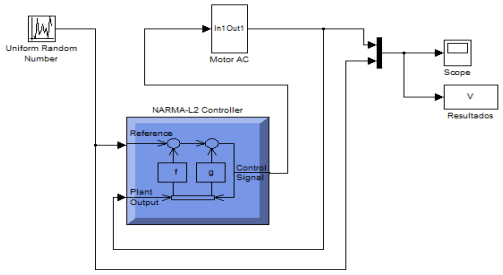


Fig. 10. Modelo del controlador Narma-L2.

Para la simulación es necesario introducir los datos en la red elegida para el control, los parámetros establecidos fueron; 14 neuronas en la capa oculta, un entrenamiento en línea de 10000 muestras, un valor máximo y mínimo de entrada de 390 y 150 muestras respectivamente, un máximo de salida de 400 muestras y un mínimo de 400 muestras.

La generación de los datos toma un tiempo considerable de aproximadamente dos horas y media, esto se debe a que la arquitectura simple de la red interna del controlador hace que el controlador requiera de largos períodos computacionales.

Aceptado los datos, una vez establecido un número total de muestras de mejoramiento en el entrenamiento de 1000 y definido el algoritmo de entrenamiento a la red Powell-Beale *conjugate gradient backpropagation*, se genera la formación de la red.

La salida de la planta y la señal de referencia se muestran en la figura 11.

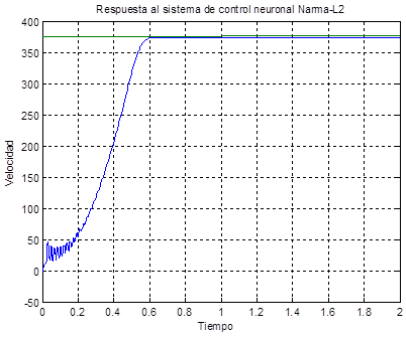


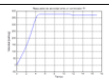
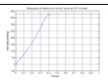
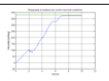
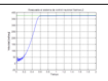
Fig. 11. Salida de la planta y señal de referencia frente al controlador Narma-L2.

Al simular el sistema en lazo cerrado utilizando Narma-L2 el sistema presenta cambios bruscos al inicio en su respuesta transitoria, producto a la inductancia mutua del motor de inducción, con pequeño error permanente en estado estacionario.

4. ANÁLISIS DE LOS RESULTADOS

Una vez realizado la simulación del sistema de control de velocidad del motor de inducción con varias estrategias de control se estableció la comparación de los resultados en la siguiente tabla.

Tabla 2: Comparación de los resultados

	CONTROL PI	PID CLONADO
Gráficas de respuesta del sistema		
Error en estado estacionario	0.15 apróx.	0 apróx.
Tiempo Respuesta transitoria	0.49 apróx.	0.42 apróx.
Etapas de diseño del controlador	• Análisis de la respuesta transitoria y estacionaria del modelo. • Determinar los valores de los parámetros del controlador.	•Identificar un modelo de red neuronal del controlador que se desea reemplazar en la planta. •el modelo neuronal toma el lugar del modelo convencional PI/PID.
Tiempo de Generación	-	5-10 min apróx.
	PREDICTIVO	**NARMA-L2**
Gráficas de respuesta del sistema		
Error en estado estacionario	3.2 apróx.	2.45 apróx.
Tiempo Respuesta transitoria	0.7 apróx.	0.6 apróx.
Etapas de diseño del controlador	•Identificación del sistema (desarrollar un modelo de red neuronal de la planta que se desea controlar). •Diseño del control (el modelo de la planta se utiliza para predecir el futuro comportamiento de la planta y una optimización del algoritmo se utiliza para seleccionar el control de entrada que optimiza el rendimiento futuro).	•Identificación del sistema (desarrollar un modelo de red neuronal de la planta que se desea controlar). •Diseño del control (el controlador es simplemente un reordenamiento del modelo de la planta).
Tiempo de Generación	150 min apróx.	240 min apróx.

El controlador PID clonado, sin sobreoscilaciones, un tiempo de respuesta corto y entrega cero como error en estado estacionario, lo que lo hace ser un sistema óptimo.

El modelo del controlador predictivo presenta cambios bruscos en su respuesta transitoria, con un rango de error superior a los otros modelos trabajados en la investigación.

El modelo del controlador neuronal Narma-L2 requiere de largos periodos computacionales. Este sistema presenta cambios bruscos al inicio de su respuesta transitoria, con error en estado estacionario. Este modelo es uno de los que presenta mejor respuesta transitoria.

5. CONCLUSIONES

La respuesta óptima del sistema, según los cuatro controladores presentados, sería el PID clonado que se presenta error cero, alcanzando una respuesta mínima a su punto de referencia. Sin embargo, por su complejidad, estructura y diseño los controladores Predictivo y Narma-L2 presenta una alta eficiencia.

El modelo de referencia, debido al tamaño que se necesita de los datos se hizo el método más difícil y fue imposible lograr su entrenamiento. La implementación de este tipo de controladores neuronales es una desventaja.

La implementación del modelo neuronal PID clonado es el recomendado de las diferentes estrategias de control empleadas en este trabajo.

Por último, se sugiere explorar con el sistema de inferencia neurodifuso ANFIS (*Adaptive Neuro-Fuzzy Inference Systems*), por sus posibilidades de integrar las mejores características de los sistemas difusos y redes neuronales. Permitiendo a la técnica ANFIS el ajuste automático de controladores difusos y modelar sistemas para predecir su comportamiento futuro.

REFERENCIAS

Pardo G., A.; Díaz R., J. L. *Aplicaciones de convertidores de frecuencia. Estrategias PWM*, Universidad de Pamplona, 2005.

Nildo, J. y Guilherme, L. *Análisis de los parámetros, modelados matemáticamente, usados en al modelación de motores*, 1995.

Sotelo, V. *Controlador de velocidad para motores AC utilizando técnicas de campo orientado y redes neuronales con un Sistema híbrido FPGA-DSP*, Perú, 2007.

Acevedo G., T. L. *Diseño e Implementación de un controlador Lógico Difuso aplicado a un Motor de Inducción*, Tesis de Maestría, Universidad de Pamplona, Pamplona, 2008.

Díaz R., J. L. *Control por Campo Orientado del Motor de Inducción con Adaptación de los Parámetros por Modelo de Referencia*, Tesis de Maestría, UCLV, Cuba, 2000.

Kung, S. Y. *Digital neural networks*, PTR Prentice Hall, Inc. 1993.

Sowilan, A. *Aplicación de las redes neuronales en los sistemas de control vectorial de los motores de inducción*, Universidad Politécnica de Cataluña, Tesis doctoral, 2000.

Fraustro, C. *Redes neuronales en el sistema de control vectorial del motor de inducción*, 2008.

Acosta, M. y Zuluaga; C. *Tutorial sobre redes neuronales aplicadas en ingeniería eléctrica y su implementación en un sitio Web*, Universidad Tecnológica de Pereira, 2000.

González, J.; Da Silveira, M. y Pacheco J. *Comparación de la red neuronal y el filtro de Kalman en la estimación de velocidad del motor de inducción*, 2004.

Mathworks. *Help Neural Network*, Matlab 7.12. PDF, 2006.

Freeman, J. y Skapura, D. *Redes Neuronales: Algoritmos, aplicaciones y técnicas de programación*. Delaware E.U.A Addison Wesley Iberoamericanas, 1993.

Delgado, A. *Propiedades matemáticas y aplicaciones de las redes neuronales dinámicas Recurrentes*. Universidad Nacional de Colombia. Santafé de Bogotá, Colombia, 1998.

Jang, J.S.R. "ANFIS: Adaptive-Network-Based Fuzzy Inference System", IEEE Trans. Systems, Man, Cybernetics, 23(5/6):665-685, 1993.