

INFORME

Ref.: GAMHE-03001

Desarrollo e implantación de un sistema de control borroso del esfuerzo de corte y su aplicación en la máquina herramienta.

AUTORES: R.E. Haber Guerra, J.R. Alique, A. Alique

Año: 2003

Resumen

El estudio que recoge este informe está dividido en dos partes. En la primera, se explora la utilización del paradigma conocido como linealización por realimentación utilizando redes neuronales artificiales (RNA) con vistas a resolver el problema del control del esfuerzo de corte en un sistema electromecánico complejo. El principal objetivo es controlar el esfuerzo de corte modificando la velocidad de avance. El funcionamiento se analiza utilizando varias cifras de mérito. Los resultados demuestran que esta estrategia proporciona un buen rechazo a las perturbaciones para los casos analizados. En la segunda parte, se aborda la utilización del control por modelo interno utilizando RNA y lógica borrosa con el objetivo de realizar el control del esfuerzo de corte en un sistema electromecánico complejo, representado en este trabajo por el proceso de mecanizado, el objetivo principal de este regulador consiste en controlar una variable de salida, el esfuerzo de corte, variando una variable de entrada, la velocidad de avance. El esquema consta de dos partes la primera es un modelo dinámico utilizando RNA para estimar la salida del proceso, y el segundo un controlador borroso con la misma ganancia estática que el modelo RNA inverso, para determinar las acciones de control necesarias (velocidades de avance) que permitan mantener constante el esfuerzo de corte. Tres aportes son analizados en esta segunda parte a saber: controlador lógico borroso (FLC), controlador por modelo interno neuronal (IMC) y controlador neuroborroso (NFC). El estudio incluye la simulación y la evaluación del comportamiento basado en diferentes cifras de mérito. Los resultados demuestran que el controlador neuroborroso proporciona un mejor rechazo a las perturbaciones que el INC y el FLC para los casos analizados.

Introducción

El desarrollo e implantación de un sistema de control borroso del esfuerzo de corte tiene sus orígenes en la propia necesidad de desarrollar un sistema inteligente de control del esfuerzo de corte y su aplicación en máquinas herramientas provistas de CNC abiertos [8, 10, 13, 15]. Durante el diseño del prototipo se han tenido en cuenta no sólo los requisitos que exige el mecanizado convencional sino también otros aspectos (por ejemplo, un tiempo de cómputo de las acciones de control menor de un milisegundo) que puedan servir de base a futuros desarrollos en procesos de mecanizado a alta velocidad [12].

De este modo el prototipo no comercializable acerca la máquina herramienta española al estado del arte en el tema del control adaptativo de los procesos de mecanizado por medio de técnicas de Inteligencia Artificial. Las investigaciones han sido dirigidas hacia todas las etapas del diseño de un sistema de control desde la medición, el análisis y procesamiento de señales hasta la investigación de la aplicabilidad industrial de las más modernas estrategias para la toma de decisiones basadas en la lógica borrosa (LB) y en el modelado basado en las redes neuronales artificiales (RNA).

El desarrollo de un prototipo no comercializable, como paso previo a su fabricación industrial, es vital ya que en estos momentos la competencia en el sector de la máquina herramienta es intensa, y a medio plazo los mercados tenderán saturarse. Sólo aquellos fabricantes que sean capaces de incorporar mejoras en las prestaciones de las máquinas que lo distinguan del resto podrán abrirse paso en un mercado globalizado. Así, los resultados trazarán el camino para hacer frente a la competencia de Estados Unidos y Japón en este sector. Ciertamente, uno de los retos de España para los venideros años consiste en consolidar su posición en el mercado globalizado, no sólo dentro del conjunto de la Unión Europea, sino como un símbolo a seguir por el resto de países de la UE menos desarrollados.

Desde el punto de vista económico, los resultados obtenidos en este proyecto permitirán a medio y corto plazo que los fabricantes españoles de máquina herramientas coloquen en el mercado una máquina que disponga de un sistema inteligente para el control del esfuerzo de corte, de fácil montaje (sin sensores), de bajo coste de instalación (software desarrollado sobre entornos clásicos) y reducidos tiempos de calibración. Como resultado del proyecto se ha obtenido un prototipo capaz de disminuir el tiempo de mecanizado en un 10%, utilizando como restricción la propia potencia de la máquina y la vida útil de la herramienta de corte, y el funcionamiento del prototipo embebido en CNC abiertos, sin sensores ni dispositivos de medición adicionales (sólo por medio de las señales internas de los accionamientos).

A pesar de los resultados positivos obtenidos con el prototipo los trabajos deben continuar con vistas a mejorar las prestaciones del controlador y mejorar el aprovechamiento del reducido tiempo disponible para la medición, procesamiento y elaboración de las acciones correctivas. Este informe recoge los últimos resultados del estudio realizados donde se exploran diferentes esquemas de control basados en RNA, LB y esquemas híbridos de control. Las dificultades existentes para realizar los ensayos en tiempo real han propiciado que los resultados mostrados en este estudio abarquen simulaciones basadas en modelos aproximados del proceso de mecanizado.

Sin embargo, cabe señalar que estos modelos, así como todos los datos utilizados en el aprendizaje y entrenamiento de las RNA son reales y adquiridos en tiempo real de la máquina-herramienta. Esto nos permite confiar y aprovechar en buena medida estos resultados tanto por parte de SORALUCE para perfeccionar sus máquinas.

Este informe está dividido en dos partes. En la primera parte se analizan los resultados de la aplicación del aporte conocido como linealización por realimentación (FBL) basado en RNA para control del esfuerzo de corte, se hace una revisión bibliográfica de los aportes más destacados en esta temática y se abordan todos los aspectos del diseño y verificación del sistema de control a lazo cerrado. En la segunda parte se analizan otros esquemas de control basados en estrategias ya conocidas como el diseño de controladores neuronales por modelo interno (IMC), el diseño de controladores borrosos (FLC) y el diseño de controladores neuro-borrosos basados en IMC.

**Parte I: Linealización por realimentación utilizando redes neuronales artificiales.
Aplicación a un proceso electromecánico complejo: el proceso de mecanizado.**

Resumen

Este estudio explora la utilización del paradigma conocido como linealización por realimentación utilizando redes neuronales artificiales (RNA) con vistas a resolver el problema del control del esfuerzo de corte en un sistema electromecánico complejo. El principal objetivo es controlar el esfuerzo de corte modificando la velocidad de avance. El funcionamiento se analiza utilizando varias cifras de mérito. Los resultados demuestran que esta estrategia proporciona un buen rechazo a las perturbaciones para los casos analizados.

1.1 Introducción

En las últimas décadas el FBL ha recibido mucha atención por parte de la comunidad científica como vía para considerar las imprecisiones en los modelos cuando se diseñan sistemas de control. La idea principal del FBL radica en encontrar una ley de control de realimentación y estática de modo que el sistema a lazo cerrado, describa un comportamiento lineal entrada-salida. El FBL exacto abarca etapas como es la transformación de un sistema no lineal en un sistema lineal, utilizando ecuaciones diferenciales de primer orden, lo que con frecuencia es bastante complicado. Por otra parte la limitación a cerca de la disponibilidad del conocimiento exacto de las no linealidades del proceso y las dificultades relativas a la transformación no lineal han motivado la utilización de la técnica FBL con las RNA [2].

Ciertamente FBL con RNA en los sistemas de control constituye un área de investigación bastante activa en la actualidad. En este sentido importantes contribuciones aparecen reportadas en la literatura para asegurar acciones de control acotadas [3]. Además se sugiere en la literatura diferentes aportes para diseñar un neurocontrolador utilizando FBL [4], no obstante el procedimiento no es tan sencillo como remarcan los autores cuando el modelo y sus parámetros son parcialmente conocidos o totalmente desconocidos, y la transformación es difícil de llevar a cabo. De igual modo se han utilizado RNA en esquemas MRAC utilizando estrategias basadas en control por modelo interno [5]. El aporte más reciente en esta temática es el FBL generalizado. A pesar de todos estos progresos es necesario desarrollar

investigaciones futuras para incorporar el rechazo a perturbaciones no medibles y la selección automática de parámetros para alcanzar la estabilidad a lazo cerrado.

Un modo de hacer frente a los sistemas electromecánicos complejos aprovechando los beneficios que ofrecen las técnicas del control lineal es identificar y cancelar las no linealidades, utilizando el paradigma FBL, en ese trabajo un proceso electromecánico complejo, el proceso de mecanizado, es utilizado como plataforma de ensayos para aplicar un aporte donde se combinan FBL y RNA [8]. La parte eléctrica de este sistema incluye motores de DC y AC, amplificadores, sensores y otros componentes, mientras que la parte mecánica incluye la estructura rígida y el cuerpo de la máquina herramienta, con los diferentes ejes, rodamientos y reductores. El principal objetivo es implementar la optimización del proceso de mecanizado controlando el esfuerzo de corte. Un neurocontrolador estático y fijo es diseñado a partir del FBL, para controlar el proceso linealizado. La efectividad del esquema propuesto se demuestra a través de simulaciones y diferentes cifras de méritos obtenidas de los resultados de la simulación.

1.2 El proceso de mecanizado

Las características del proceso de mecanizado considerado como un sistema electromecánico complejo, limita extraordinariamente la utilización de herramientas matemáticas clásicas para el modelado y el control, la dinámica del proceso de fresado puede ser modelada de forma aproximada, utilizando una ecuación diferencial de segundo orden. El siguiente modelo ha sido obtenido a partir de datos experimentales entrada-salida, obtenidos en la fresadora SL-400, y guarda mucha similitud con el modelo lineal sugerido en [9].

$$G_{LU}(z) = \frac{0.019z + 0.017}{z^2 - 1.75z + 0.77} \quad (1)$$

Otro modelo obtenido utilizando ecuaciones diferenciales de segundo orden [10] es.

$$G_{RS}(z) = \frac{0.052z + 0.04}{z^2 - 1.42z + 0.45} \quad (2)$$

Por otra parte modelos de primer orden del esfuerzo de corte incluyendo no linealidades en la profundidad de corte y en las velocidades de corte se propone en [11].

$$G_{LA}(z) = \frac{0.11}{z - 0.85} a^{0.65} f^{0.63}. \quad (3)$$

Donde $G_P = \{G_{LU}(z), G_{RS}(z), G_{LA}(z)\}$ representa el proceso de mecanizado desde un punto de vista clásico. Las ecuaciones (1)-(3) son solamente válidas en un rango reducido de modo que no pueden sobrepasar ciertos límites en la representación de la complejidad e incertidumbre inherentes del proceso de mecanizado. No obstante proporcionan una información muy valiosa a cerca del comportamiento dinámico por medio de una caracterización aproximada. De ahí que sean utilizados en la sección 1.5 para las simulaciones.



Figura 1. Vista general de la máquina herramienta SL-400

1.3 RNA. Preliminares

Consideremos un sistema lineal con $\mathbf{u}_k \in R^m$ entradas y $\mathbf{y}_k \in R^p$ ($m \geq p$) salidas. Asumamos que el sistema puede ser modelado de forma exacta por la siguiente RNA de una sola capa.

$$y_{k+1} = \mathbf{W}_o \cdot \tanh(\mathbf{W}_x \cdot \mathbf{x}_k + \mathbf{W}_u \cdot u_k + \mathbf{b}_x) + \mathbf{b}_o \quad (4)$$

Donde $\mathbf{x}_k \in R^n$ es dada por

$$\mathbf{x}_k = [y_k^T, y_{k-1}^T, \dots, y_{k-ny}^T, u_{k-1}^T, \dots, u_{k-nu}^T]^T \quad (5)$$

con $n = (ny + 1) \cdot p + nu \cdot m$, $\mathbf{W}_o \in R^{p \times l}$, $\mathbf{W}_x \in R^{l \times n}$, $\mathbf{W}_u \in R^{l \times m}$, $\mathbf{b}_x \in R^{l \times 1}$ y $\mathbf{b}_o \in R^{p \times 1}$, donde \mathbf{b}_x es el vector de polarización en la capa interna, \mathbf{b}_o es el vector de polarización en la capa de salida, \mathbf{W}_x , \mathbf{W}_u son las matrices de pesos y \tanh representa la función tangente hiperbólica.

Por motivos de sencillez consideremos un sistema no lineal

SISO ($y_k \in R^1$). La identificación se puede ver como el procedimiento para determinar el mapeo desde un conjunto $z^N = [\mathbf{u} \ \mathbf{y}]^T$ a un conjunto de posibles pesos (parámetros) $\hat{\boldsymbol{\theta}}$ de modo que la RNA pueda hacer una predicción \hat{y}_{k+1} tan cerca como sea posible de la salida real del proceso y_{k+1} .

$$\hat{\boldsymbol{\theta}} = [\text{vec}(\hat{\mathbf{W}}_o) \ \text{vec}(\hat{\mathbf{W}}_x) \ \text{vec}(\hat{\mathbf{W}}_u) \ \hat{\mathbf{b}}_x \ \hat{\mathbf{b}}_o]^T \quad (6)$$

Utilizando el método conocido como identificación basada en la predicción del error [12], los pesos son calculados como

$$\hat{\boldsymbol{\theta}} = \text{argmin}(E(\boldsymbol{\theta}, z^N)). \quad (7)$$

Una versión del método de Levenberg-Marquardt fue seleccionada en el algoritmo de entrenamiento [13].

El primer paso en la utilización de FBL es identificar el sistema que va a ser controlado. De las dos estructuras o esquemas generales propuestos en [14] podemos seleccionar el modelo serie paralelo, mostrado en (8), que expresa el proceso no lineal a través de una función $g(\cdot)$ siendo un modelo NARMA

$$\begin{aligned} \hat{y}(k+1|\boldsymbol{\theta}) &= g(\boldsymbol{\varphi}(k), \hat{\boldsymbol{\theta}}) \\ &= g(y_k, y_{k-1}, \dots, y_{k-ny+1}, u_k, \dots, u_{k-nu+1}, \hat{\boldsymbol{\theta}}) \end{aligned} \quad (8)$$

Utilizando (4) para formar el modelo serie paralelo de (8), el modelo puede ser escrito nuevamente como (9) y (10).

$$\phi_{k(q)} = \tanh\left(\sum_{i=1}^{my} a_{i(q)} y_{k-i+1} + \sum_{j=1}^{nu} b_{j(q)} u_{k-j+1} + b_{x(q,0)}\right) \quad (9)$$

$$\hat{y}_{k+1} = \sum_{q=1}^p W_{o(k,q)} \cdot \phi_{k(q)} + b_{o(k,0)} \quad (10)$$

Con el objetivo de aplicar el FBL entrada –salida que se explicará en la próxima sección, es necesario utilizar modelos *afines* (modelos donde las entradas tienen que aparecer linealmente en la descripción en espacio de estados del modelo). El modelo descrito por (10) no es afín ya que las entradas no aparecen linealmente en la salida. Considerando una aproximación del modelo (8) se obtiene (11)

$$\begin{aligned} y_{k+1} &= f_n(y_k, y_{k-1}, \dots, y_{k-ny+1}, u_{k-1}, \dots, u_{k-nu+1}) + \dots \\ &\quad g_n(y_k, y_{k-1}, \dots, y_{k-ny+1}, u_{k-1}, \dots, u_{k-nu+1}) \cdot u_k \end{aligned} \quad (11)$$

La ecuación (11) es homóloga de la ecuación (10) de una forma compacta. Ahora el asunto principal consiste en entrenar dos redes neuronales $f_n(\cdot), g_n(\cdot)$, con el mismo número de entradas para obtener un modelo *afine*, de la dinámica del sistema. El método L-M puede ser utilizado para determinar los pesos de $f_n(\cdot), g_n(\cdot)$. La derivada de la predicción con respecto a los pesos $\psi(k|\theta) = \frac{\partial \hat{y}(k|\theta)}{\partial \theta}$ es la componente principal en la implementación de este método

de entrenamiento, así como la Hessiana (segunda derivada) $\frac{\partial^2 \hat{y}(k|\theta)}{\partial \theta^2}$.

$$\frac{\partial \hat{y}(k|\theta)}{\partial \theta} = \begin{bmatrix} \frac{\partial \hat{f}_n}{\partial \theta_{f_n}} \\ \frac{\partial \hat{g}_n}{\partial \theta_{g_n}} \cdot u(k-1) \end{bmatrix}. \quad (12)$$

Una vez más consideremos una RNA con una capa oculta y funciones de activación tangente hiperbólica en la capa oculta y lineal en la capa de salida. La derivada de la salida del modelo con respecto a los pesos consiste en la derivada de cada red.

$\frac{\partial \hat{f}_n}{\partial \theta_{f_n}}$ se puede calcular utilizando (13)-(14)

$$\frac{\partial \hat{f}_n(k|\theta_{f_n})}{\partial W_{o(i,q)}} = \begin{cases} \phi_{k(q)}, & q > 0, k = i \\ 1, & q = 0, k = i \\ 0, & i \neq k \end{cases} \quad (13)$$

$$\frac{\partial \hat{f}_n(k|\theta_{f_n})}{\partial W_{x(i,q)}} = \begin{cases} W_{x(k,q)} \cdot (1 - \phi_{k(q)}^2) \cdot \phi_{k(j)}, & j > 0 \\ W_{x(k,q)} \cdot (1 - \phi_{k(q)}^2), & j = 0 \end{cases} \quad (14)$$

$$\frac{\partial \hat{f}_n(k|\theta_{f_n})}{\partial W_{u(i,q)}} = \begin{cases} W_{u(k,q)} \cdot (1 - \phi_{k(q)}^2) \cdot \phi_{k(j)}, & j > 0 \\ W_{u(k,q)} \cdot (1 - \phi_{k(q)}^2), & j = 0 \end{cases}$$

1.4 FBL con RNA

Primero la RNA es entrenada para aprender la dinámica del proceso, siendo conocidos un conjunto de datos entrada-salida el entrenamiento en nuestro caso se llevó a cabo utilizando datos obtenidos en tiempo real a partir de operaciones reales de mecanizado. De acuerdo con (8) la ecuación dinámica puede escribirse de forma reducida por:

$$\hat{F}(t) = g(\mathbf{F}, \mathbf{v}). \quad (15)$$

Donde g es la función desconocida (modelo) a identificar, F es el esfuerzo de corte necesario para arrancar las virutas y v es la velocidad relativa entre la herramienta y la bancada. Expresado de forma vectorial \mathbf{v} y \mathbf{F} son la entrada y la salida definidas como $\mathbf{F} = [F(k-1) \ \dots \ F(k-n)]$ y $\mathbf{v} = [v(k-1), \ \dots, v(k-m)]$, k es el instante de tiempo discreto. Un buen resultado de la identificación debe asegurar valores de $\hat{F}(k)$ cercanos a la salida real del proceso $F(k)$.

El algoritmo de entrenamiento fue desarrollado utilizando Matlab y datos en tiempo real de operaciones de mecanizado. La topología seleccionada inicialmente fue de una entrada v y una salida \hat{F} (sistema SISO), función de activación lineal en la salida y una capa interna con función de activación tangente hiperbólica. El tipo de modelo (orden) fue seleccionado a partir del conocimiento a priori del proceso y los tipos de modelos considerados en trabajos anteriores (nuestros y de otros autores). Una red neuronal con seis neuronas en la entrada seis en la capa interna y una en la capa de salida fue seleccionada. La ecuación dinámica se puede describir en una notación reducida por

$$\hat{F}(k+1) = g(v(k), F(k), F(k-1)) . \quad (16)$$

La utilización de FBL con RNA se puede comprender de forma sencilla, utilizando un ejemplo de un sistema SISO. Consideremos que se conoce que el modelo discreto de un proceso es de segundo orden. Con vistas a aplicar la teoría de FBL son necesarios modelos *afines*

$$\hat{F}_{k+1} = f_n(\hat{F}_k, \hat{F}_{k-1}, v_{k-1}) + g_n(\hat{F}_k, \hat{F}_{k-1}, v_{k-1}) \cdot v_k . \quad (17)$$

Después que se obtienen $f_n(\cdot)$, $g_n(\cdot)$ a partir de la metodología descrita en la sección 1.3 se obtiene (síntesis) un controlador por FBL calculando la acción de control a partir de (17)

$$v_k = \frac{F_r(k) - f_n(\hat{F}_k, \hat{F}_{k-1}, v_{k-1})}{g_n(\hat{F}_k, \hat{F}_{k-1}, v_{k-1})} . \quad (18)$$

Seleccionando F_r como combinaciones lineales de salidas anteriores más la referencia nos permite, realizar una asignación fija de los polos del sistema a lazo cerrado. El comportamiento entrada salida (lineal), será por tanto igual al del siguiente sistema lineal de segundo orden.

$$F_r(k) = 1.6 \cdot F_r(k-1) - 0.64 \cdot F_r(k-2) + 0.04 \cdot r(k) . \quad (19)$$

Los resultados del modelo (15) para predecir en tiempo real el modelo de corte se muestran en la figura 2. Diferentes criterios fueron utilizados para evaluar la calidad del modelo. El *NSSE* fue 0.37 el *FPE* 0.39 y el *ENV* fue 0.77.

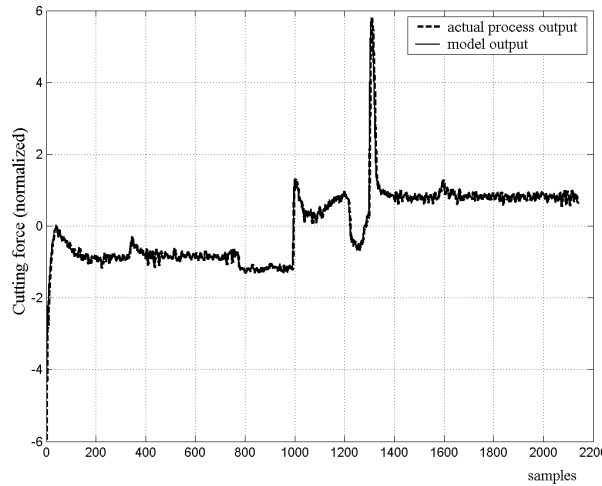


Figura 2. Comparación entre la salida real y la estimada por el modelo

1.5 Simulaciones y resultados

En lo que sigue las simulaciones están basadas en los modelos (1)-(3) que representan modelos aproximados del proceso y en la utilización del esquema de control representado en la figura 3.

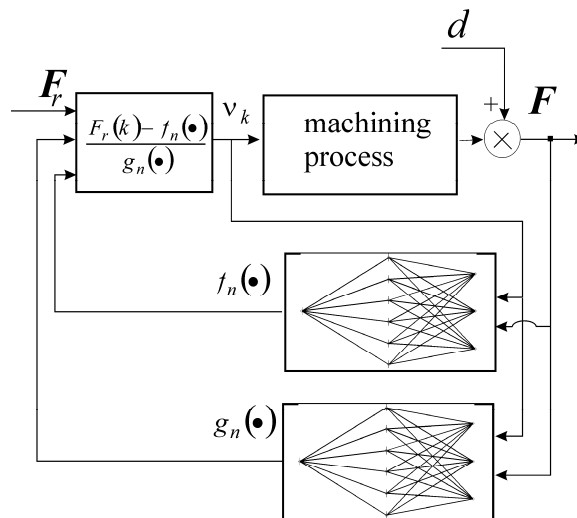


Figura 3. Esquema del FBL utilizando RNA.

Con vistas a analizar la capacidad de rechazo ante las perturbaciones del sistema de control, se ha considerado ruido aditivo mas la influencia de dinámicas no modeladas con vistas a acercarnos al comportamiento esperado del sistema físico real. El siguiente ruido aditivo que actúa a la salida

$$d(t) = 0.1 \cdot (\sin 8t + \sin 12t + \sin 23.66t + \sin 35.49t) . \quad (20)$$

Ahora el modelo realista del proceso incluye dinámicas no modeladas multiplicativas y (18) lo que se representa por

$$G_p(z) = G_*(z) \cdot \frac{0.095}{z - 0.904} + d(z) . \quad (21)$$

Donde $G_*(z)$ es el modelo ideal del proceso representado por (1) –(3).

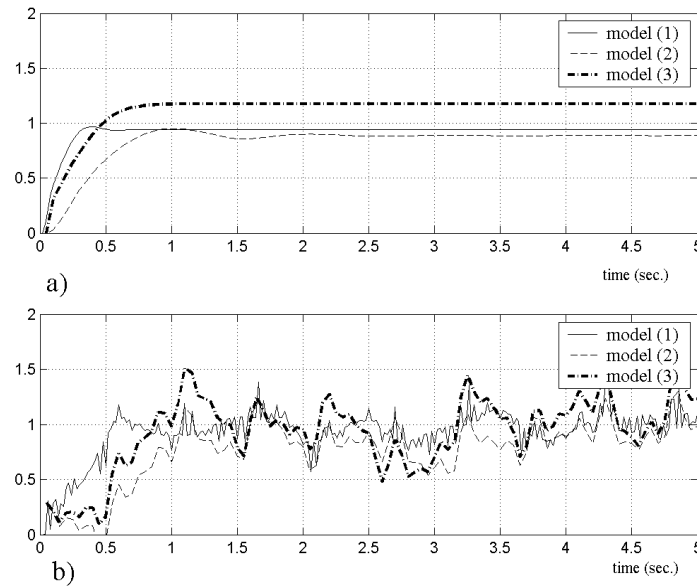


Figura 4. Respuesta a lazo cerrado sin y con ruido.

La figura 4 muestra los resultados de la simulación sin y con la influencia de dinámicas no modeladas y perturbaciones. En la tabla 1 se muestran los resultados de las diferentes cifras de méritos utilizadas para evaluar el funcionamiento del esquema propuesto.

Modelo	ISE	IAE	ITAE	M_{pl}
(1)	4.85	19.95	36.64	-
(2)	6.88	22.54	65.08	-
(3)	6.57	28.16	100.23	17.72
(21)-(1)	12.96	35.68	63.40	38.61
(21)-(2)	15.22	33.15	81.92	30.01
(21)-(3)	14.97	35.45	115.81	56.61

Tabla 1. Resumen de las cifras de mérito evaluadas

Conclusiones

Este trabajo presenta los primeros resultados de una estrategia de control basada en el paradigma FBL, que establece una sinergia muy útil entre una RNA entrenada a partir de datos reales y un esquema FBL tradicional sin necesidad de una invertibilidad estable. Los resultados aunque preliminares son positivos de modo que esta estrategia nos permite estabilizar el proceso en presencia de ruido.

Las tareas subsiguientes del proyecto CONBOMAQ estarán encaminadas hacia la aplicación en tiempo real en un CNC abierto Sinumerik 840-D del esquema propuesto y de las redes neuronales obtenidas.

Parte II. Control no lineal por modelo interno (IMC) utilizando redes neuronales (RNA) y lógica borrosa (FL) .

Resumen

Este estudio aborda la utilización del control por modelo interno utilizando RNA y lógica borrosa con el objetivo de realizar el control del esfuerzo de corte en un sistema electromecánico complejo [9,11, 14,16], representado en este trabajo por el proceso de mecanizado, el objetivo principal de este regulador consiste en controlar una variable de salida, el esfuerzo de corte, variando una variable de entrada, la velocidad de avance. El esquema consta de dos partes la primera es un modelo dinámico utilizando RNA para estimar la salida del proceso, y el segundo un controlador borroso con la misma ganancia estática que el modelo RNA inverso, para determinar las acciones de control necesarias (velocidades de avance) que permitan mantener constante el esfuerzo de corte. Tres aportes son analizados en esta segunda parte a saber: controlador lógico borroso (FLC), controlador por modelo interno neuronal (IMC) y controlador neuroborroso (NFC). El estudio incluye la simulación y la evaluación del comportamiento basado en diferentes cifras de mérito. Los resultados demuestran que el controlador neuroborroso proporciona un mejor rechazo a las perturbaciones que el INC y el FLC para los casos analizados.

2.1 Introducción

En la actualidad las RNA y la FL son probablemente las técnicas de inteligencia artificial más ampliamente utilizadas en la identificación (obtención de modelos) y el diseño de sistemas de control [24]. Las RNA han mostrado una excelente capacidad para representar cualquier función no lineal con el grado de precisión deseado. Debido a esta característica las RNA son adecuadas para la identificación y control de sistemas no lineales [25, 26]. De todas las clases de redes las *feedforward* y en particular el perceptrón multicapa son las más frecuentemente utilizadas en el control no lineal. Igualmente, la FL ha mostrado su utilidad en la ingeniería de control como una herramienta muy práctica en la optimización de procesos complejos. El número creciente de aplicaciones industriales que en el ámbito mundial han tenido éxito constituye la mejor credencial de esta sólida técnica [27], y es que a través de la FL podemos incorporar a los sistemas de control conocimiento humano expresado en

términos cualitativos, proporcionándonos una herramienta matemática poderosa para el cómputo y la integración de información cualitativa y cuantitativa.

La combinación de diferentes técnicas de inteligencia artificial en diferentes campos del control automático ha producido aplicaciones industriales eficientes e interesantes [28]. Este trabajo apunta hacia esa misma dirección y muestra resultados preliminares de una estrategia de control sinérgica entre una RNA dinámica entrenada a partir de datos reales y utilizada para estimar la salida del proceso y un controlador borroso que mejora el funcionamiento global del sistema a lazo cerrado.

El control por modelo interno (IMC) es un aporte bien fecundo y establecido en la literatura para diseñar controladores de modo que el modelo del proceso es utilizado de forma explícita (síntesis directa) en el procedimiento de diseño de un controlador. [29, 30], no obstante cuando no está disponible el modelo matemático del proceso o este es muy sofisticado, el IMC puede valerse de un modelo explícito obtenido por identificación experimental (conocido como aporte “caja negra”). En tales casos las RNA pueden utilizarse para obtener un esquema de control eficiente basado en el paradigma IMC. El IMC basado en RNA consiste en primer lugar en entrenar una red para aprender la dinámica del proceso. Otra RNA puede ser entrenada para aprender la dinámica inversa del proceso tal que pueda ser utilizada como un controlador no lineal [13]. Si nosotros podemos sintetizar un controlador borroso FLC con una ganancia estática igual a la ganancia estática del modelo inverso, y si el sistema de control es estable con este controlador, entonces podemos obtener (sintetizar) un controlador para valores de referencia constantes y capaz de hacer frente a las perturbaciones aditivas en la salida [31]. El esquema de control neuroborroso que se propone en este trabajo está precisamente inspirado en el IMC – RNA, de manera que el modelo inverso RNA se sustituye en el camino directo por el controlador borroso, apuntando al mejoramiento de la capacidad de rechazo al ruido.

La efectividad del esquema FLC se demuestra a través de simulaciones. La comparación entre los tres esquemas se realiza, teniendo como base varios índices de comportamiento.

2.2 Aporte basado en FLC para la regulación del esfuerzo de corte

El diseño de un controlador borroso que aparece detallado en [] se introduce en esta sección las variables de entrada son el error en el esfuerzo de corte ΔF y la deriva del error en el esfuerzo de corte $\Delta^2 F$, la variable manipulada que se ha seleccionado es

el incremento en la velocidad de avance Δv mientras que la velocidad de giro se considera constante y fijada por el operador, el esquema de control se muestra en la figura 5 la salida del controlador se infiere a partir de la regla composicional. El operador *sup-producto* fue seleccionado para la regla composicional de inferencia. Por ejemplo aplicando la norma T_2 (producto) y aplicando la conorma S_1 (máximo) se obtiene (22)

$$\mu(\Delta F, \Delta^2 F, \Delta v) = S_1 \left[T_2 \left[\mu_{\Delta F_i}(\Delta F), \mu_{\Delta^2 F_i}(\Delta^2 F), \mu_{\Delta v_i}(\Delta v) \right] \right] \quad (22)$$

Donde T_2 representa la operación del producto algebraico y S_1 representa el operador unión (máximo) $m \times n = 49$ reglas. La salida del controlador utilizada para cambiar la velocidad de avance se obtiene por desemborronado utilizando el método del centro del área (COA)

$$\Delta v = \frac{\sum_i \mu_R(\Delta v_i) \cdot \Delta v_i}{\sum_i \mu_R(\Delta v_i)} \quad (23)$$

La estrategia utilizada para calcular v determina que tipo de regulador borroso va a ser utilizado, en este caso es un regulador borroso PI:

$$v^*(k) = v(k-1) + \Delta v(k) \quad (24)$$

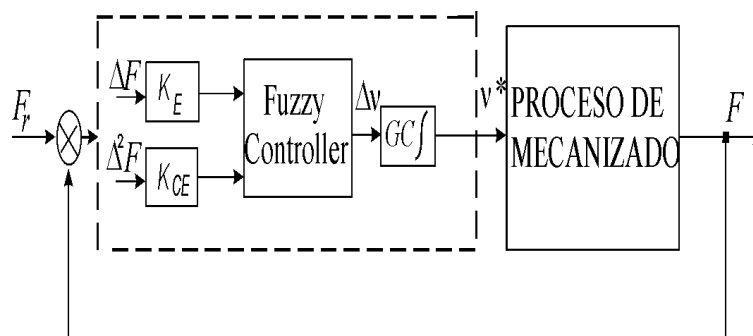


Figura 5. Esquema de control borroso para el mecanizado

2.3 Sistema IMC basado en redes neuronales

Primeramente, la RNA es entrenada para aprender la dinámica del proceso, y por tanto son conocidos los conjuntos de datos entrada-salida. El entrenamiento se realiza utilizando datos en tiempo real obtenidos de operaciones reales de mecanizado. Uno

de los modelos de redes neuronales desarrollados en [26] se selecciona como la base para el IMC.

El algoritmo de entrenamiento fue desarrollado en Matlab. La topología inicialmente escogida por una entrada y una salida función de activación lineal en la salida y una capa interna con función de activación tangente hiperbólica, el tipo de modelo se seleccionó a partir del conocimiento ingenieril del proceso y otros tipos de modelos reportados en la literatura. La RNA tuvo una estructura 4-4-1 y la ecuación dinámica en notación reducida viene dada por

$$\hat{F}(t+1) = G_M(v(t) \quad v(t-1) \quad F(t) \quad F(t-1)) . \quad (25)$$

Donde $G_M = \{w_f, W_f\}$

$$w_f = \begin{bmatrix} -0.4355 & 0.0901 & 0.7938 & -0.5570 & -0.4723 \\ 0.3790 & 0.3895 & -0.2630 & -0.1079 & 0.1944 \\ 0.5686 & -0.3144 & -0.7093 & 0.8608 & 0.1641 \\ -0.2621 & 0.7771 & -0.3636 & 0.4915 & 0.9354 \end{bmatrix} \quad (26)$$

$$W_f = [-0.6189 \quad 0.2987 \quad 1.0114 \quad 1.1728 \quad -1.2777] . \quad (27)$$

El modelo inverso fue obtenido a partir de la técnica conocida como entrenamiento generalizado [13]. Por tanto la red se entrenó *off-line* para minimizar

$$J(\theta) = \sum_{t=1}^M (f(t) - \hat{f}(t))^2 . \quad (28)$$

Otra red neuronal se entreno para aprender la dinámica inversa del proceso y trabajar como un controlador no lineal. La ecuación dinámica del modelo inverso en notación reducida queda expresada como:

$$f(t) = G'_M(F(t+1) \quad F(t) \quad f(t-1) \quad f(t-2)) . \quad (29)$$

Donde $G'_M = \{w_i, W_i\}$

$$w_i = \begin{bmatrix} 0.1020 & -0.0190 & -0.0096 & -0.1468 & 0.0670 \\ 1.5430 & 0.6541 & -0.2669 & -0.7483 & 0.2376 \\ 0.1216 & -0.1247 & 0.0590 & 0.2472 & -1.1422 \\ 0.1328 & -0.0265 & -0.0356 & 0.1701 & 0.5677 \end{bmatrix} . \quad (30)$$

$$W_i = [-3.3957 \quad 0.0569 \quad 2.0374 \quad 2.7758 \quad 0.4598] . \quad (31)$$

El vector de polarización esta representado por la última columna de las matrices (25)-(27) y (29) – (31), los resultados del modelo directo en la predicción en tiempo real del esfuerzo de corte se muestran en la figura 6a. El comportamiento del modelo inverso se muestra en la figura 6b.

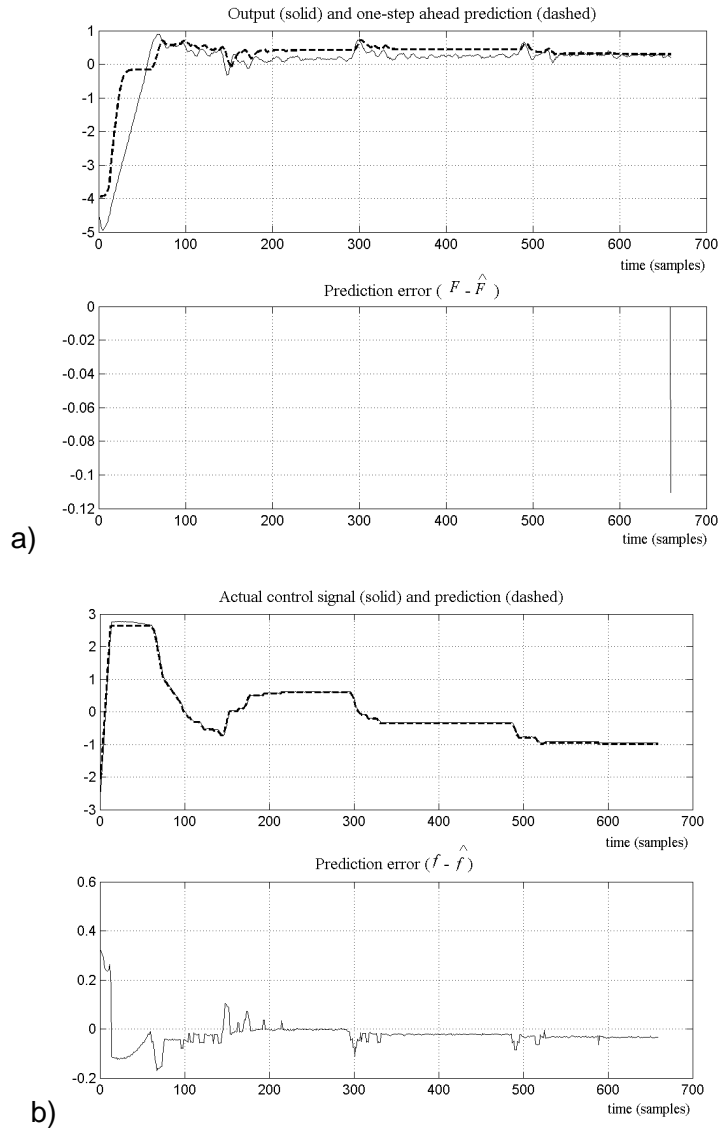


Figura 6. Comportamiento en tiempo real a) del modelo directo b) modelo inverso

Se utilizaron los mismos índices de comportamiento descritos en la sección 1.4 para evaluar el modelo directo y el modelo inverso definidos como

$$NSSE = \sqrt{\frac{SSE}{2N}} \quad (32)$$

$$FPE = \frac{NSSE(N + p)}{N - p} \quad (33)$$

$$ENV = \frac{2 \cdot N \cdot NSSE}{N - p} \quad (34)$$

$$SBC = N \cdot \log\left(\frac{SSE}{N}\right) + p \cdot \log(N) \quad (35)$$

Donde SSE es la suma del cuadrado de los errores, $NSSE$ es el error en la predicción, FPE es el error final en la predicción y p es el número efectivo de pesos.

Se obtuvo para el modelo directo $NSSE_f=0.37$, $FPE_f=0.39$, $ENV_f=0.77$ y $SBC_f=-3942.7$, mientras que para el modelo inverso se obtuvo $NSSE_i=0.0274$, $FPE_i=0.0291$, $ENV_i=0.0564$ y $SBC_i=-2831.2$.

2.4 Simulaciones y resultados

Las simulaciones se realizaron utilizando los modelos del proceso (1) – (3) y los esquemas de control representados en la figura 5 y 7.

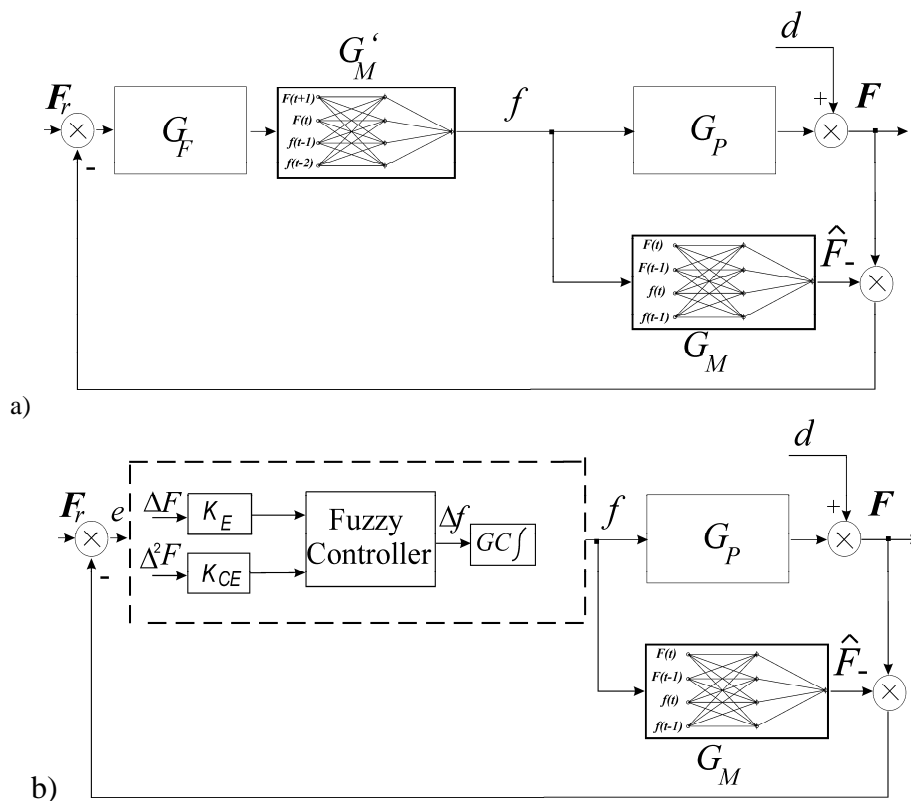


Figura 7 a) Diagrama del IMC, b) esquema del NFC.

El filtro G_F se incluye con el objetivo de reducir la ganancia de alta frecuencia y mejorar la robustez del sistema. Este filtro también suaviza los cambios rápidos y bruscos en las señales mejorando la respuesta transitoria del controlador. Por tanto la

robustez está considerada de forma explícita en el diseño por modelo interno, aunque puede ser ajustada escogiendo adecuadamente el filtro G_F .

Una forma bastante común del filtro es su selección como un filtro paso bajo. Utilizando la transformada Z el filtro $G_F(z)$ quedaría como

$$G_F(z) = \frac{1 - k_2}{z - k_1} \quad (36)$$

Donde k_1, k_2 son los parámetros de diseño, y $k_1 = k_2$.

Los resultados del estudio comparativo de las prestaciones del FLC IMC y NFC se resumen en la tabla 2. Se utilizaron los mismos criterios que en la sección 1.5. Tanto el FLC como el IMC son capaces de regular el proceso en presencia de ruido. El FLC ofrece una respuesta transitoria ligeramente mejor que el IMC pero con un mayor error y un mayor sobrepaso. El NFC funciona adecuadamente tanto en el transitorio como ha estado estacionario.

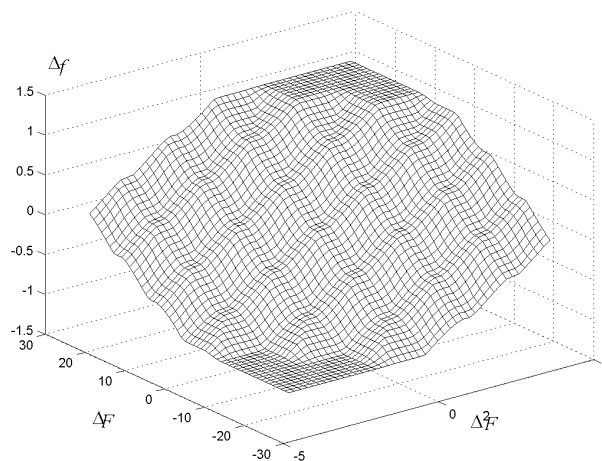
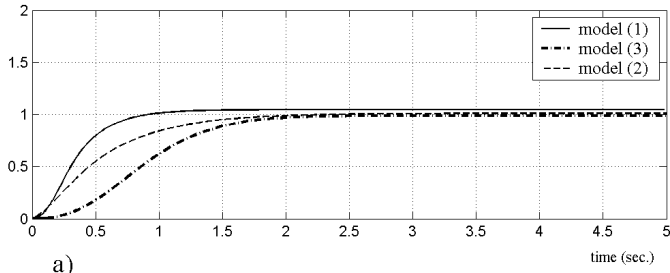
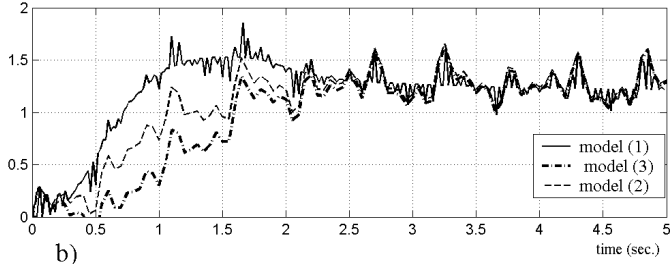


Figura 8. Característica estática del NFC.

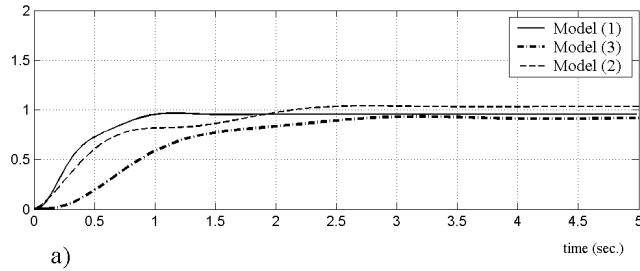


a)

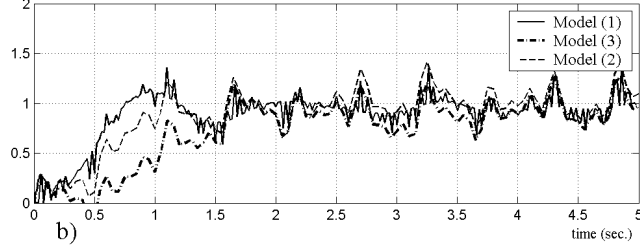


b)

Figura 9. Comportamiento utilizando FLC

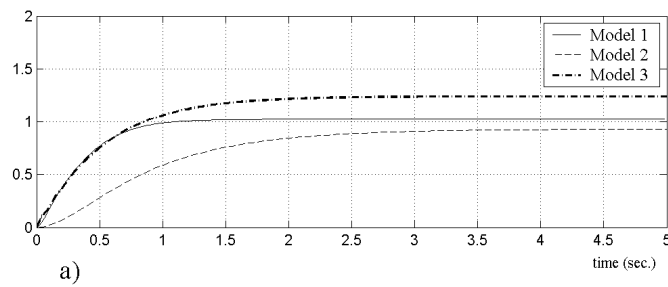


a)

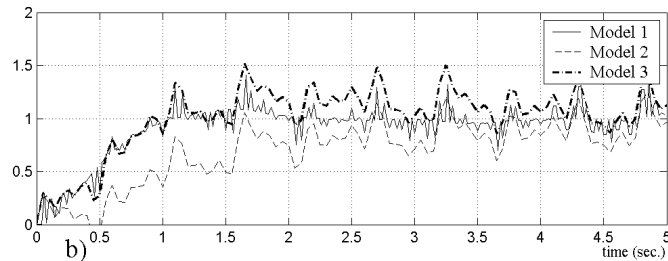


b)

Figura 10. Comportamiento utilizando IMC



a)



b)

Figura 11. Comportamiento utilizando NFC

Criterio	ISE			IAE			ITAE			M_{pt} [%]		
	FLC	IMC	NFC	FLC	IMC	NFC	FLC	IMC	NFC	FLC	IMC	NFC
(1)	12.4	20.0	10.5	21.9	28.5	21.5	12.7	31.7	18.1	NO	NO	2.41
(2)	13.9	15.4	13.8	20.2	30.4	29.0	16.9	57.4	53.4	NO	NO	NO
(3)	7.1	7.1	11.1	13.0	16.1	36.3	10.8	23.8	130.3	1.2	4.0	23.9
(21)-(1)	43.3	20.7	18.4	90.2	43.4	38.0	180.8	64.7	50.9	85.2	35.9	41.2
(21)-(2)	25.6	20.4	19.8	48.7	38.2	39.4	136.6	88.0	90.1	61.4	27.7	27.3
(21)-(3)	20.6	11.9	11.1	44.6	27.5	27.9	140.1	66.6	73.0	65.3	41.3	51.6

Tabla 2. Comparación de las diferentes estrategias de control.

Conclusiones

Este trabajo presenta algunos resultados preliminares de una estrategia de control basada en el paradigma del diseño por modelo interno, siendo una estrategia sinérgica de una red neuronal de una RNA entrenada a partir de datos reales para predecir la salida del proceso y un controlador borroso que mejora el funcionamiento del sistema de control en su conjunto. El modelo inverso representado por una RNA se sustituye con vistas a mejorar la capacidad de rechazo a las perturbaciones.

Los resultados, aunque todavía preliminares, muestran como el NFC funciona mejor que el FLC y el IMC a partir de los diferentes criterios de evaluación utilizados. No obstante la sencillez del FLC y lo satisfactorio de su respuesta transitoria puede convertir a este esquema en una buena solución para hacer frente a los estrictos requisitos y especificaciones de tiempo de computo que exige el mecanizado a alta velocidad

Referencias

1. Khalil, H., Nonlinear Control Systems, Prentice Hall, 2nd Edition, (1996)
2. Hagan, M.T., Demuth, H.B., de Jesus, O.: An Introduction to the Use of Neural Networks in Control Systems. International Journal of Robust and Nonlinear Control **12** (2002) 959-985
3. Yesildrek, A., Lewis, F.L.: Feedback Linearization Using Neural Networks. Automatica **31** (1), (1995) 1659-1664
4. He, S., Reif, K., Unbehauen, R.: A Neural Approach for Control of Nonlinear Systems with Feedback Linearization. IEEE Transactions of Neural Networks **9** (6) (1998) 1409-1421

5. Lightbody, G., Irwin, G.W.: Nonlinear Control Structures Based on Embedded Neural Systems Models. *IEEE Transactions on Neural Networks* **8**(3) (1997) 553-567
6. Goodwin, G.C., Rojas, O., Takata, H.: Nonlinear Control via Generalized Feedback Linearization Using Neural Networks. *Asian Journal of Control* **3**(2) (2001) 79-88
7. Buckner, G.D., Schuetze, K.T., Beno, J.H.: Intelligent Feedback Linearization for Active Vehicle Suspension Control: *Journal of Dynamic Systems, Measurement and Control* **123** (2001) 727-733.
8. R. Haber, J. Alique, S. Ros, and C. R. Peres, "Fuzzy supervisory control of end milling process," *Information sciences*, vol. 89, no. 1-2, pp. 95-106, 1996.
9. Alique A., R. E. Haber, R. H. Haber, S. Ros, and C. Gonzalez, "A neural network-based model for the prediction of cutting force in milling process. A progress study on a real case," in *Intelligent Control, 2000. Proceedings of the 2000 IEEE International Symposium on*, 2000, pp. 121-125: IEEE.
10. Haber R. E., R. Haber, A. Alique, and S. Ros, "Hierarchical fuzzy control of the milling process with a self-tuning algorithm," in *Intelligent Control, 2000. Proceedings of the 2000 IEEE International Symposium on*, 2000, pp. 115-120: IEEE.
11. Haber R. E., R. Haber, C. Peres, A. Alique, and S. Ros, "Model of the milling process on the basis of cutting force: A Neural network approach," in *Advances in Neural Networks and Applications (N. Mastorakis Ed.)*, Artificial Intelligence Series, World Scientific and Engineering, 2001.
12. Haber R., A. Alique, J. Alique, R. Haber-Haber, and S. Ros, "Current trends and future developments of new control systems based on fuzzy logic and their application to high speed machining," *Revista de Metalurgia(Spain)*, vol. 38, no. 2, pp. 124-133, 2002.
13. Haber R. E., A. Alique, S. Ros, and R. Haber, "Application of knowledge-based systems for supervision and control of machining processes," in *Handbook of Software Engineering and Knowledge Engineering: Volume II: Emerging Technologies*, 2002, pp. 673-709.
14. Haber R. E., R. Haber, A. Alique, S. Ros, and J. R. Alique, "Dynamic model of the machining process on the basis of neural networks: from simulation to real time application," in *International Conference on Computational Science*, 2002, pp. 574-583: Springer, Berlin, Heidelberg.
15. Haber R. E., J. R. Alique, A. Alique, J. Hernández, and R. Uribe-Etxebarria, "Embedded fuzzy-control system for machining processes: Results of a case study," *Computers in Industry*, vol. 50, no. 3, pp. 353-366, 2003.
16. Haber, R.E., Haber, R.H., Ros, S., Alique, A., Alique, J.R.: Dynamic Model of the Machining Process on the Basis of Neural Networks: From Simulation to Real Time Application. *Lecture Notes in Computer Science* **2331** (2002) 574-583
17. Lauderbaugh, L.K., Ulsoy, A.G.: Model Reference Adaptive Force Control in Milling. *ASME Journal Engineering of Industry* **111** (1989) 13-21
18. Rober, S.J., Shin, Y.C.: Control of Cutting Force for Milling Processes Using an Extended Model Reference Adaptive Control Scheme. *Journal of Manufacturing Science and Engineering* **118** (1996) 339-347
19. Landers, R., Ulsoy, A.: Model-Based Machining Control. *ASME Journal of Dynamics Systems, Measurement and Control* **122**(3) (2000) 521-527
20. Ljung, L.: *System Identification: Theory for the User*: 2nd Edition, Prentice Hall, (1999)

21. Norgard, M., Ravn, O., Poulsen, N.K., Hansen, L.K.: Neural Networks for Modelling and Control of Dynamic Systems, Springer-Verlag, London (2000)
22. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. *IEEE Transactions on Neural Networks* **1** (1990) 4-27.
23. Narendra, K.S., Mukhopadhyay, S.: Adaptive Control Using Neural Networks and Approximate Models. *IEEE Transactions on Neural Networks* **8** (1997) 475-485.
24. Hornik K., Stinchcombe M., White H.: Multilayer feedforward networks are universal approximators. *Neural Networks* **2** (1989) 359-366
25. Hunt K.J., Sbarbaro D., Zbikowski R., Gawthrop P.J.: Neural Networks for control systems- A survey. *Automatica* **28**(6) (1992) 1083-1112
26. Nelles O.: Nonlinear system identification: From classical approaches to neural networks and fuzzy logic, Springer Verlag (2000)
27. Hung S.J., Lian R.J.: A hybrid fuzzy logic and neural networks algorithm for robot motion control. *IEEE Transactions on Industrial Electronics* **44**(3) (2000) 217-229.
28. Morari M., Zafiriou E.: Robust process control. Prentice Hall, Englewood Cliffs, NJ, (1989).
29. Goodwin G.C., Graebe S.F., Salgado M.E.: Control System Design. Prentice Hall, NJ, USA, pp. 567-582 (2001).
30. Rivals I., Personnaz L.: Nonlinear internal model control using neural networks: Application to processes with delay and design issues. *IEEE Transactions on Neural Networks* **11**(1) (2000) 80-90.