

CONTROLADOR PREDICTIVO NEURO-GENÉTICO

Alberto Aguado, Alfredo Gómez, Abelardo del Pozo

*Instituto de Cibernética, Matemática y Física, Calle 15 No. 551,
entre C y D, Vedado,
La Habana. e-mail: aguado@icmf.inf.cu*

Resumen: En este trabajo, se presenta una solución al problema del control predictivo basado en modelos, que se distingue por utilizar un modelo neuronal como predictor y un algoritmo genético, especialmente diseñado para ser aplicado en tiempo real, como optimizador. Este enfoque es aplicable al control de procesos fuertemente no lineales que no pueden ser aproximados eficientemente por algún modelo lineal y posibilita, además, el uso de criterios de optimización arbitrarios, no necesariamente cuadráticos y la inclusión de soluciones heurísticas ad hoc para mejorar los resultados. En el trabajo, se presentan algunos ejemplos preliminares del uso del controlador propuesto, obtenidos en régimen de simulación. *Copyright © 2007 CEA-IFAC*

Palabras clave: Control predictivo, algoritmos genéticos, redes neuronales, control de unidades caldera-turbina

1. INTRODUCCIÓN

A partir de los trabajos pioneros de Clarke y otros (1987a, 1987b), el Controlador Predictivo Generalizado (GPC por su nombre en inglés) y muchos otros métodos semejantes, del llamado control predictivo basado en modelo, han sido tratados profusamente en la literatura y han encontrado un espacio considerable en numerosas aplicaciones industriales (Camacho y Bordons, 2004).

Muchos de los métodos más populares de este tipo de control predictivo tienen en común, entre otros aspectos, la utilización de un modelo lineal como predictor y la de un criterio cuadrático como función objetivo a minimizar.

Aunque este esquema funciona en la mayoría de los casos prácticos, es indudable que adolece de una limitación excluyente cuando, o bien un modelo lineal no es capaz de funcionar como un predictor eficiente, o cuando por alguna razón resulta conveniente utilizar un criterio no cuadrático, o incluso, no analítico.

La utilización de modelos neuronales y/o algoritmos genéticos en esquemas de control predictivo aparece ampliamente reportada en la literatura. Sin pretender hacer un análisis exhaustivo, a continuación mencionamos algunos trabajos relevantes en este sentido.

En el trabajo de Martínez et al. (1998) se propone un esquema de control predictivo en el que participa un algoritmo genético como optimizador, lo que posibilita utilizar criterios no cuadráticos. No obstante en este esquema se mantiene el uso de predictores lineales tal como se hace en los esquemas clásicos.

En Zamarreño y Vega (1999) se propone una metodología consistente en la modelación de un sistema fuertemente no lineal mediante una red neuronal recurrente y el uso de un método de optimización no-lineal del tipo quasi-newton. En este trabajo se utiliza un criterio cuadrático convencional como función objetivo para la optimización y se incluyen restricciones en las variables de control y controlada.

En el trabajo de Shin y Park (1998), así como en Aguado et al (1999) se proponen métodos de control predictivo basados en modelos neuronales y optimizadores genéticos. No obstante, en estos trabajos, no se tienen en cuenta las dificultades de implementación en tiempo real, cuando el espacio de búsqueda del control óptimo es definido en forma convencional.

En el presente trabajo, se retoma el tema del control predictivo neuro-genético (NG), introduciendo algunas innovaciones importantes: en el algoritmo genético utilizado, en la estructura del modelo neuronal y en el tratamiento de las restricciones en el control. Se cuenta, además, con las mejoras tecnológicas de las computadoras (mucho mayor velocidad) y las posibilidades incrementadas de los ambientes de simulación actualmente disponibles.

Quizás la novedad más importante que se propone en este trabajo es que los individuos de la población que se procesa en el algoritmo genético, se definen como vectores de incrementos en el control. Esto permite utilizar poblaciones pequeñas e introducir algunas reglas heurísticas que contribuyen a mejorar la calidad del control, como se verá en los ejemplos de simulación propuestos.

Los resultados preliminares, obtenidos en régimen de simulación, parecen indicar que la metodología propuesta puede dejar de ser solamente una posibilidad teórica para convertirse en una interesante posibilidad práctica.

El trabajo está organizado como sigue: En el epígrafe 2, se explica el modelo neuronal propuesto y una metodología para su entrenamiento. En 3, se explican las características del algoritmo genético utilizado. En 4 se propone la estructura del controlador predictivo neuro-genético. Por último, en 5 y 6 se muestran algunos ejemplos de su comportamiento en régimen de simulación y en 7 se ofrecen algunas conclusiones.

2. MODELO NEURONAL

Consideremos el siguiente modelo neuronal, cuya estructura puede considerarse como la de un modelo ARX generalizado (Aguado, 2000):

$$y(t) = N(x(t)) + \lambda(t) \quad (1)$$

donde $x(t)$ es el vector de entrada a la red neuronal que se define como:

$$x(t) = [u(t-1) \ y(t-1) \ \dots \ u(t-n) \ y(t-n) \ 1] \quad (2)$$

El último término 1 se incluye para tener en cuenta la desviación del modelo neuronal. Por

simplicidad, aunque sin pérdida de generalidad, se asume un orden único del modelo n , de manera que la dimensión del vector de entrada será $2n+1$.

El término $\lambda(t)$ en la ecuación (1) representa el error del modelo, que puede tener diversas fuentes: perturbaciones no medibles en el proceso, un orden del modelo inadecuado, entrenamiento deficiente de la red neuronal o insuficiencia de los datos, etc. Como la mejor aproximación posible, se supone que $\lambda(t)$ es un proceso estocástico de incrementos independientes, o sea:

$$\lambda(t) = \lambda(t-1) + \xi(t) \quad (3)$$

donde $\xi(t)$ se considera un proceso de ruido blanco estacionario, cuya esperanza matemática es cero. Tenemos entonces que:

$$\begin{aligned} E\{\lambda(t+1)/x(t)\} &= E\{\lambda(t+2)/x(t)\} = \dots \\ &\dots = E\{\lambda(t+N_2)/x(t)\} = \hat{\lambda}(t) \end{aligned} \quad (4)$$

donde N_2 es el horizonte máximo de predicciones futuras de la salida, que serán calculadas utilizando el modelo neuronal.

También:

$$\hat{\lambda}(t) = y(t) - N(x(t)) \quad (5)$$

es la estimación del error que puede hacerse en el tiempo t , con los datos disponibles.

Las predicciones futuras de la salida, cuyo cálculo será necesario en el esquema de control predictivo, pueden calcularse entonces mediante el Algoritmo I, que se ofrece más abajo. La estructura de la red neuronal utilizada como modelo tiene algunas particularidades que a continuación se detallan:

- Se trata de un perceptrón que puede entrenarse mediante un algoritmo de retropropagación convencional.

Algoritmo I

Datos iniciales:

$y(t)$, $x(t)$, $\hat{\lambda}(t)$, secuencia de valores futuros del control: $u(t)$, $u(t+1)$, \dots , $u(t+N_u)$, $N_u \leq N_2$

Comienzo:

Para $i=1:N_2$
 Si $i > N_u$
 $u(t+i) = u(t+N_u)$
 fin

$$x(t+i) = [u(t+i-1) \hat{y}(t+i-1) \\ x(t+i-1)(1:2n-2) \ 1]$$

$$\hat{y}(t+i) = N(x(t+i)) + \hat{\lambda}(t)$$

fin

donde $\hat{y}(t) = y(t)$.

- La dimensión del vector de entrada como ya vimos es $2n+1$, siendo n el orden supuesto para el modelo.
- Puede tener una o más capas escondidas, aunque en los ejemplos estudiados hasta el momento, una capa ha sido suficiente.
- La función de excitación de las neuronas de la(s) capa(s) escondida(s) se define como una función sigma.
- La capa de salida consta de una sola neurona cuya función de excitación es lineal, lo cual facilita el entrenamiento y evita considerar problemas engorrosos de escalaje.
- Se propone que el entrenamiento de la red se haga a partir de una muestra de datos de entrada y salida que, por consideraciones prácticas, debe ser lo más pequeña posible. Esta se reciclará en un número grande de épocas de entrenamiento, hasta obtener la aproximación requerida.

Para ilustrar el método de entrenamiento así como los resultados que pueden obtenerse, a continuación se presenta un ejemplo consistente en encontrar un modelo neuronal suficientemente aproximado de un sistema fuertemente no lineal. Consideremos, en efecto, el sistema:

$$y(t) = 0.8y(t-1) + 0.1\sin(y(t-2)) + 2.5u(t-1)^2 \\ + 0.8u(t-2)^3 \quad (6)$$

Para este sistema se propone un modelo neuronal aproximado con las siguientes características:

Orden del modelo neuronal: $n = 2$

Neuronas en la capa de entrada: $N_{ent} = 5$

Neuronas en la capa escondida: $N_{escon} = 4$

Tamaño de la muestra de datos: $N_{muest} = 300$

Datos para entrenamiento: $N_{entren} = 240$

Datos para validación: $N_{val} = 60$

Función de activación de las neuronas de la capa escondida: sigmoide

Función de activación de la neurona de salida: lineal

La muestra de datos para el entrenamiento y la validación del modelo se obtuvo aplicando a la entrada del sistema una señal aleatoria uniformemente distribuida en el intervalo $(-1,1)$.

Para el entrenamiento de la red se selecciona el algoritmo de retropropagación, que es el más simple posible y que en general conduce a resultados satisfactorios. La simplicidad del algoritmo y el tamaño relativamente pequeño de la muestra de datos se pagan, no obstante, con la necesidad de realizar un número elevado de épocas de entrenamiento.

Otro aspecto importante es la selección adecuada del coeficiente de aprendizaje η . Se ha comprobado experimentalmente que la calidad del entrenamiento se incrementa considerablemente si se sigue el procedimiento de disminuir los valores de η en la medida en que el error medio cuadrático de la salida de la red se va haciendo más pequeño.

En el presente ejemplo, se ha utilizado un total de 1200 épocas de entrenamiento, organizadas en 40 grupos de 30 épocas. Después de concluido un grupo de épocas, se examina la disminución del error medio cuadrático y se aplica la siguiente regla:

$$\text{Si } \text{abs}(e^2m(i) - e^2m(i-1)) < 0.015 * e^2m(i) \\ \text{entonces } \eta(i) = 0.8 * \eta(i-1) \quad (7) \\ \eta(0) = 0.02$$

donde:

$$e^2m(i) = \frac{1}{N_{muest}} \sum_{k=1}^{N_{muest}} (y(k) - y_n^i(k))^2 \quad (8)$$

es el error medio cuadrático después de concluido el grupo i de épocas de entrenamiento. $y(k)$ y $y_n^i(k)$ son respectivamente los datos de salida y las salidas de la red neuronal.

Los valores numéricos de los parámetros de la regla (7) utilizados en el presente ejemplo, fueron determinados por prueba y error sin mayores dificultades; resulta claro que en otros casos será necesario aplicar un procedimiento semejante.

En las figuras 1, 2 y 3 se muestran respectivamente las salidas superpuestas de la red y del sistema, la evolución del error medio cuadrático y la del coeficiente de aprendizaje. Como se comprobará más adelante, para los fines del control predictivo, el entrenamiento del modelo neuronal es totalmente satisfactorio.

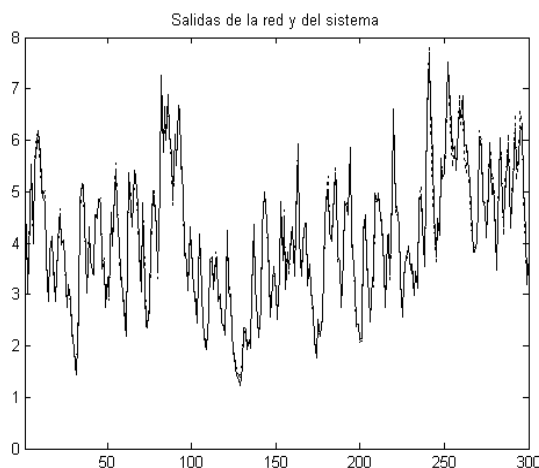


Figura 1. Salidas del sistema y del modelo neuronal superpuestas

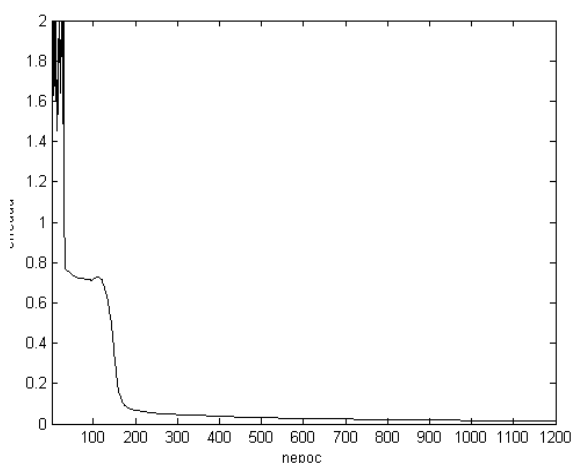


Figura 2. Evolución del error medio cuadrático con las épocas de entrenamiento

3. OPTIMIZACIÓN MEDIANTE ALGORITMOS GENÉTICOS

Un algoritmo genético (AG) es una técnica de optimización que realiza la búsqueda de la solución del problema de optimización imitando los mecanismos de evolución de las especies. En este tipo de algoritmos, un conjunto de individuos (población) cambia de generación en generación, sufriendo un proceso de evolución y adaptándose cada vez mejor al entorno. En un problema de optimización, disponemos de una función a optimizar (maximizar o minimizar) y una zona donde buscar el óptimo (espacio de búsqueda), de manera que a cada punto del espacio de búsqueda le corresponde un valor de la función objetivo y la meta es encontrar el punto que optimice (minimice o maximice) dicha función.

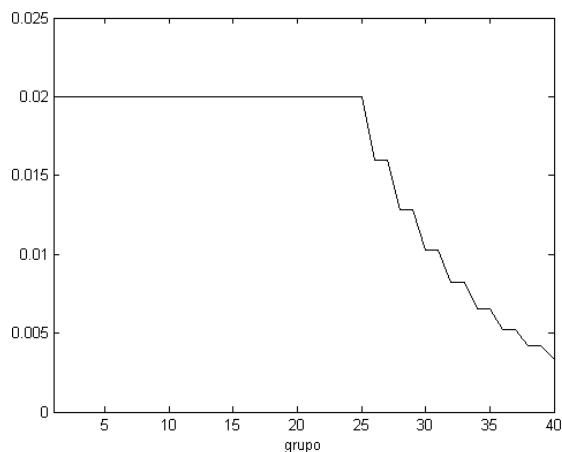


Figura 3. Evolución del coeficiente η con los grupos de épocas

Trasladado el problema de optimización a un AG, los distintos puntos del espacio de búsqueda se identifican con los posibles individuos distintos de una población.

Como ocurre con la genética natural, cada individuo se caracteriza por sus cromosomas. En el problema de optimización, el cromosoma de un individuo está constituido por las coordenadas del punto en el espacio de búsqueda:

$$z = (z_1, z_2, \dots, z_n)$$

Siguiendo con el símil, cada una de las coordenadas corresponde a un gen. El valor de la función objetivo de un individuo (un punto del espacio) debe interpretarse como el grado de adaptación al entorno de dicho individuo. Por ejemplo, si el problema consiste en encontrar el mínimo de una función $J(z)$, se entiende que un individuo está mejor adaptado al entorno que otro si el valor de la función para ese individuo es menor. Una vez establecidas, por una parte, las relaciones entre individuos (sus cromosomas) y los puntos del espacio de búsqueda, y por otra parte, el grado de adaptación al entorno y la función a minimizar, queda por describir cómo se produce la evolución de una población. Para ello se plantean distintos tipos de operaciones sobre la población, lo que se conoce como operadores genéticos. El mecanismo general de funcionamiento de un AG se podría resumir como sigue:

A partir de una población inicial (normalmente generada de forma aleatoria), se siguen los siguientes pasos:

- Se seleccionan unos individuos para la siguiente generación. La selección se realiza en función de su nivel de adaptación (valor de su función de costo). Aquellos individuos que tienen un

valor $J(z)$ más bajo, tienen mayor probabilidad de ser elegidos.

- Se producen cruces entre individuos de la nueva generación que consisten en un intercambio de genes entre cromosomas (es decir, intercambio de coordenadas entre puntos). La tasa de individuos que se cruzan se ajusta con la probabilidad de cruce P_c . Se trata de mezclar los cromosomas de dos individuos de manera que se dispone de un mecanismo para explorar puntos diferentes del espacio. Se considera que este tipo de exploración del espacio de búsqueda es orientada; se basa en la información de los individuos padres y lo que se pretende es extraer las cualidades potenciales que ya existen en la población.

- Algunos de los individuos de la nueva generación sufren mutaciones que consisten en variaciones aleatorias de los genes (es decir, variaciones aleatorias de las coordenadas de un punto). La probabilidad de mutación se ajusta con el parámetro P_m . Al igual que las operaciones de cruce, la mutación permite explorar zonas diferentes del espacio de búsqueda. Sin embargo, esta exploración no orientada es totalmente aleatoria, pretende desarrollar la capacidad de exploración de zonas que no pueden ser exploradas por los individuos existentes en la población actual. En este marco general, aparecen diversas variantes en la implementación de los algoritmos genéticos; variando el tipo de codificación de los genes, variando los mecanismos de selección, cruce o mutación, incorporando nuevos operadores genéticos, etc. La implementación que se ha realizado para resolver el problema de control predictivo es semejante a la utilizada en el trabajo de Martínez et al. (2006) y tiene las siguientes características:

- La codificación se realiza con números reales, cada gen tiene un valor real.
- La función objetivo utilizada para la operación de selección no es directamente $J(z)$. Se realiza una operación de 'ranking' (Blasco, 1999) (Back, 1996) que consiste en ordenar de mayor a menor valor los individuos según su valor $J(z)$. Se cambia el valor de $J(z)$ por su posición en dicha clasificación, cada individuo tiene un nuevo valor $J'(z)$ que corresponde a su número de orden. Con esta operación se consigue que los individuos claramente dominantes no predominen demasiado rápido en la población, agotando prematuramente el algoritmo (este aspecto es especialmente importante en el caso

de poblaciones con pocos individuos, como es el caso del control predictivo).

- La operación de selección que se realiza se conoce como Muestreo Universal Estocástico (*Stochastic Universal Sampling (SUS)*) (Baker, 1987) de manera que la probabilidad de supervivencia de un individuo viene dada por la ecuación:

$$P(z_i) = \frac{J'(z_i)}{\sum_{j=1}^{N_{ind}} J'(z_j)} \quad (9)$$

donde N_{ind} es el número de individuos de la población.

- Para el cruce de los cromosomas se utiliza el operador conocido como recombinación intermedia que consiste en lo siguiente: Los individuos hijos z'_1 y z'_2 se obtienen realizando la siguiente operación sobre los individuos padres

$$\begin{aligned} z'_1 &= \alpha_1 z_1 + (1 - \alpha_1) z_2 \\ z'_2 &= \alpha_2 z_2 + (1 - \alpha_2) z_1 \\ \alpha_1 &\in [-d, 1 + d] \\ \alpha_2 &\in [-d, 1 + d] \end{aligned} \quad (10)$$

- La mutación de un cromosoma, seleccionado aleatoriamente con probabilidad P_m , se realiza calculando un máximo del 20% de cambio en todos sus genes, mediante la expresión:

$$z_i' = z_i (1 + 0.2 \text{randn}) \quad (11)$$

donde randn representa un número aleatorio gaussiano de media cero y varianza 1.

4. CONTROL PREDICTIVO NEURO-GENÉTICO (NG)

Vamos a considerar a continuación un método de control predictivo neuro-genético (NG) que, como su nombre lo indica, utiliza un modelo neuronal como predictor y un algoritmo genético (AG) para minimizar un criterio que expresa el comportamiento deseado del control.

Gracias a la generalidad de los AG, podemos utilizar un criterio de optimización prácticamente arbitrario, sin las restricciones que imponen otros métodos. Vamos a considerar, por ejemplo, los siguientes funcionales como criterios posibles para la tarea de optimización:

$$1) \quad J_1(t) = \sum_{k=N_1}^{N_2} (r(t+k) - \hat{y}(t+k))^2 \quad (12)$$

$$2) \quad J_2(t) = \sum_{k=N_1}^{N_2} \text{Abs}(r(t+k) - \hat{y}(t+k)) \quad (13)$$

$$3) \quad J_3(t) = \sum_{k=N_1}^{N_2} \sqrt{\text{Abs}(r(t+k) - \hat{y}(t+k))} \quad (14)$$

En los tres casos, N_1 y N_2 representan los conocidos horizonte mínimo y máximo de predicción; N_1 se selecciona teniendo en cuenta el retardo puro presumible en el sistema a controlar y para la selección de N_2 se tiene en cuenta su dinámica: para sistemas más rápidos N_2 puede ser pequeño y viceversa. $\hat{y}(t+k)$ simboliza a las predicciones de la salida futura que pueden calcularse mediante el modelo neuronal y $r(t+k)$ representa a la evolución futura de la referencia, supuestamente conocida.

Algunos comentarios sobre los criterios formulados mediante (12), (13) y (14):

- Nótese que en ninguno de ellos se incluye una penalización explícita de la variable de control, o de sus incrementos, como se hace en los métodos tradicionales de control predictivo. La razón es que en los AG, las restricciones del control se logran más eficientemente, como se verá en detalle, restringiendo adecuadamente el espacio de búsqueda.
- El funcional definido en (12) representa un criterio cuadrático tradicional que penaliza muy fuertemente los errores grandes de la variable controlada y en forma débil a los errores pequeños. Debido a esta característica, puede ocurrir en algunos casos, que la minimización de este criterio conduzca a sistemas en los que el control actúa muy enérgicamente cuando el error es grande y en forma excesivamente lenta, cuando el error se acerca a cero.
- El criterio (13), que incluye la función valor absoluto, no se utiliza en los métodos convencionales de control predictivo, debido a que no es derivable y por tanto no puede ser minimizado en forma analítica. Este criterio, sin embargo, puede ser ventajoso en algunos casos ya que los errores pequeños y grandes son penalizados en forma proporcional, de manera que el comportamiento del control puede ser más uniforme y los errores

pequeños pueden ser eliminados más rápidamente.

- El criterio (14), por otra parte, tiene un comportamiento inverso al cuadrático: los errores grandes son penalizados débilmente y viceversa, lo que puede ser conveniente en algunos casos.

La posibilidad de utilizar diversos criterios incluyendo combinaciones de varios, es una de las ventajas de utilizar los AG como método de optimización. Por ejemplo, es posible formular un criterio combinado en la forma:

Si $\text{Abs}(r(t) - y(t)) > \varepsilon$

$$J_4(t) = J_1(t) \quad (15)$$

de lo contrario,

$$J_4(t) = J_2(t)$$

donde ε es un valor pequeño seleccionado convenientemente. El criterio $J_4(t)$ eventualmente puede combinar las ventajas de los criterios $J_1(t)$ y $J_2(t)$ y puede ser minimizado mediante un AG prácticamente sin un costo adicional de cómputo.

Para la minimización del criterio seleccionado, utilizaremos un AG con las características definidas en el epígrafe 3, para el que un individuo de la población se define como:

$$z_i(t) = (\Delta u_i(t) \Delta u_i(t+1) \dots \Delta u_i(t+N_u-1)) \quad (16)$$

$$i = 1, 2, \dots, N_{ind}$$

donde los $\Delta u_i(\cdot)$ representan los incrementos del control en los tiempos sucesivos hasta $t+N_u-1$. N_u es el horizonte de control sujeto a la restricción $N_u \leq N_2$. N_{ind} es el número de individuos de la población.

Para definir el espacio de búsqueda del AG, se establecen restricciones sobre los incrementos del control del tipo:

$$-\delta \leq \Delta u_i(t+k) \leq \delta$$

$$k = 0, 1, \dots, N_u - 1 \quad (17)$$

$$i = 1, 2, \dots, N_{ind}$$

Como se apreciará más adelante, el comportamiento del controlador es muy sensible a la selección de δ : a valores grandes de este parámetro corresponden respuestas más rápidas pero eventualmente oscilatorias, mientras que su disminución generalmente conduce a respuestas más lentas.

Es posible y además, necesario desde un punto de vista práctico, incluir restricciones en el valor total de la variable de control de la forma:

$$u_{\min} \leq u(t-1) + \sum_{k=0}^j \Delta u_i(t+k) \leq u_{\max}$$

$$j = 1 \dots N_u \quad (18)$$

$$i = 1 \dots N_{ind}$$

También, sin un costo excesivo de cómputo, es posible, en los casos que sea conveniente, incluir restricciones explícitas en la variable controlada tal como:

$$y_{\min} \leq \hat{y}_{n_i}(t+k) \leq y_{\max}$$

$$k = N_1, \dots, N_2 \quad (19)$$

donde $\hat{y}_{n_i}(\cdot)$ son las predicciones que se calculan mediante el modelo neuronal. N_1 y N_2 son los horizontes mínimo y máximo de predicción.

La determinación de la secuencia de control óptima, en el horizonte de tiempo especificado, utilizando un modelo neuronal como predictor y un AG, se resuelve en cada etapa de tiempo, utilizando el algoritmo II.

A continuación ofrecemos algunas explicaciones sobre el Algoritmo II:

- El operador “rand” genera un número aleatorio, con distribución uniforme en el intervalo $[0,1]$, de manera que los individuos z de la población son vectores de dimensión N_u , cuyos componentes se encuentran en el intervalo $[-\delta, \delta]$.
- y_n es el vector de predicciones de la salida, es decir:

$$y_n = [\hat{y}(t) \ \hat{y}(t+1) \ \dots \ \hat{y}(t+N_2)] \quad (20)$$
- N_{gener} y N_{ind} representan respectivamente, el número de generaciones y el de individuos de la población utilizados en el AG.

Algoritmo II

Datos iniciales:

$y(t), u(t-1), \delta, r(t)$

$x(t) = [u(t-1) \ y(t-1) \ \dots \ u(t-n) \ y(t-n) \ 1]$

Comienzo:

$imin = 0$

$Jmin = 10^6$

$xx = x(t)$

$y_n(1) = N(x(t))$

$\hat{\lambda}(t) = y(t) - y_n(1)$

$y_n(1) = y(t)$

para $i = 1: N_{ind}$

 para $k = 1: N_u$

$z(i,k) = 2\delta(\text{rand} - 0.5)$

 fin

fin

```

para k = 1: Ngener
  para i = 1: Nind
    up = u(t-1)
    para j = 2: N2+1
      si j ≤ Nu .
        deltau = z(i, j-1)
      si no
        deltau = 0
      fin
      up = up + deltau
      si u_min > up > u_max
        ir a 1
      fin
      xx = [up y_n(j-1) xx(1: nent -3) 1]
      y_n(j) = N(xx) + λ̂(t)
      si y_min > y_n(j) > y_max
        ir a 1
      fin
    fin j
    J = J(y_n, r(t))
    si J < Jmin
      Jmin = J
      imin = i
    fin
  1 fin i
  z = AG(z)
fin k
si imin > 0
  u(t) = u(t-1) + z(imin, 1)
si no
  u(t) = u(t-1)
fin

```

- $N(xx)$ es un operador que calcula la salida del modelo neuronal con entrada xx , $AG(z)$ simboliza el procesamiento mediante el algoritmo genético de la población z .
- En el algoritmo se incluyen chequeos de límite para las variables de control y controlada, de tal forma que cualquier individuo de la población es desechado si viola alguna de las restricciones.
- Aunque eventualmente, mediante la operación de mutación puede generarse algún incremento del control que esté fuera del entorno prefijado, esto no influye significativamente sobre el comportamiento del algoritmo, pues de todas formas funcionan los límites para la variable de control total.
- Nótese que, de acuerdo con el algoritmo, se utiliza solamente el primer elemento de la secuencia óptima de control, contenida en $z(imin, 1: N_u)$, lo que corresponde a la estrategia de “horizonte móvil (Camacho y Bordons, 2004).

5. CONTROL NG DE SISTEMA DISCRETO NO LINEAL

5.1 Control de sistema sin ruido ni retardo.

A fin de mostrar las cualidades del algoritmo de control predictivo neuro-genético propuesto, se utilizará, como primer ejemplo, el control del sistema no lineal descrito en la ecuación (6).

A continuación se analizará el comportamiento del mismo sistema no lineal controlado mediante un controlador predictivo neuro-genético, que utiliza como predictor al modelo neuronal discutido en el epígrafe 2 y cuyo entrenamiento se describe mediante las figuras 1, 2 y 3.

El AG utilizado en el controlador fue discutido en los epígrafes 3 y 4; los parámetros seleccionados fueron los siguientes:

$$(21) \quad \begin{aligned} N_{ind} &= 10 \\ P_c &= 0.9 \\ P_m &= 0.4 \\ \text{No. de generaciones} &= 3 \\ \delta &= 0.01 \\ u_{min} &= 0 \\ u_{max} &= 1 \\ y_{min} &= 0 \\ y_{max} &= 15 \end{aligned}$$

Por último, los parámetros propios del controlador predictivo fueron:

$$(22) \quad \begin{aligned} N_1 &= 1 \\ N_2 &= 10 \\ N_u &= 2 \end{aligned}$$

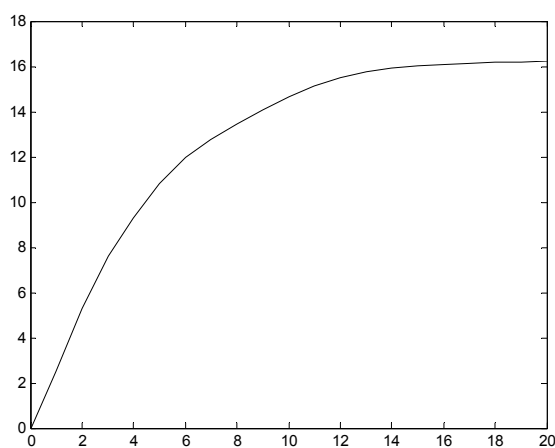


Figura 4. Respuesta del sistema no lineal ante escalón de amplitud 1

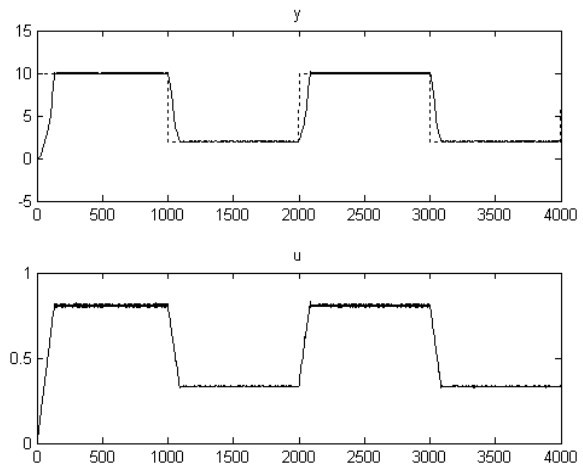


Figura 5. Sistema controlado mediante un regulador NG con criterio cuadrático

El parámetro N_2 fue seleccionado teniendo en cuenta que la respuesta del sistema ante un escalón aplicado en la entrada alcanza más del 90% del nuevo valor estacionario después de 10 unidades de tiempo, tal como se observa en la figura 4. Por otra parte, no se observó ninguna mejora significativa utilizando un horizonte de predicción mayor. En cuanto al horizonte de control N_u , no se obtuvieron soluciones mejores para valores mayores de este parámetro.

La figura 5 muestra una realización típica del controlador NG en la que se observa un seguimiento casi perfecto de la referencia mediante el controlador predictivo neuro-genético con los parámetros definidos anteriormente. También, en la segunda parte, se observa el comportamiento del control.

El criterio utilizado fue del tipo cuadrático, tal como se presenta en la ecuación (12). Los otros criterios considerados en nuestro estudio, no ofrecieron resultados diferentes en este caso. Aunque se fijaron restricciones en los valores mínimo y máximo de las variables, éstas no se activaron.

Nótese que el AG diseñado utiliza un tamaño de población y un número de generaciones muy reducidos, lo que garantiza que el esfuerzo de cómputo sea moderado.

5.2 Control del sistema no lineal con salida contaminada con ruido.

En la figura 6, se presenta el caso en que la variable controlada está contaminada con un ruido relativamente intenso y uniformemente distribuido en el intervalo $[-0.4, 0.4]$ y el criterio utilizado es el combinado definido en (15) con el valor de ε

igual a 1. Puede observarse que incluso en estas condiciones adversas, se logra mantener el control en niveles aceptables.

En este caso pueden apreciarse algunas diferencias cuando se usan los diferentes criterios definidos en las ecuaciones (12) a (15), como se observa en la Tabla 1.

Los resultados que se presentan en esta tabla se obtuvieron promediando 20 realizaciones del AG para cada criterio, lo cual compensa en parte su naturaleza aleatoria. σ_y representa al valor calculado de la desviación estándar del error de regulación después de 4000 períodos de control.

Como se observa en la Tabla 1, mientras que los criterios 2 y 3 ofrecen sólo una leve mejoría con respecto al criterio convencional cuadrático, el uso del criterio combinado $J_4(t)$ resulta francamente ventajoso en este ejemplo.

Tabla 1. Comparación de criterios en el caso de ruido en las mediciones

Criterio	σ_y
$J_1(t)$	0.4625
$J_2(t)$	0.4607
$J_3(t)$	0.4590
$J_4(t)$	0.3566

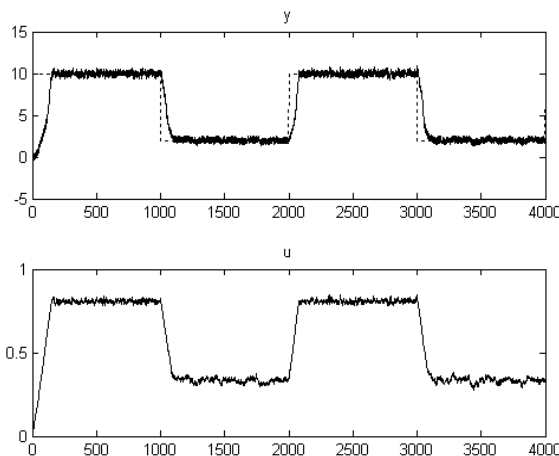


Figura 6. Sistema controlado mediante un regulador NG y ruido superpuesto en la salida. Criterio J_4

5.3 Control del sistema no lineal con retardo

Vamos a considerar el caso de la presencia de un retardo puro de 10 unidades de tiempo en el sistema no lineal y discreto definido en la ecuación 6. En este caso se entrenó un nuevo modelo neuronal, utilizando el procedimiento descrito en el epígrafe 2, con resultados similares en cuanto a

error de predicción. Los parámetros del control predictivo se seleccionaron de la forma siguiente:

$$\begin{aligned} N_1 &= 11 \\ N_2 &= 20 \\ N_u &= 2 \end{aligned} \quad (23)$$

Los parámetros del algoritmo genético se mantuvieron iguales a los que aparecen en (21).

En la figura 7, se observa la respuesta excesivamente oscilatoria que se obtiene cuando se adiciona al sistema no lineal controlado un retardo puro de 10 segundos y se mantiene el tamaño del espacio de búsqueda $\delta = 0.01$. Un intento de mejorar ese comportamiento se muestra en la figura 8, en la que el valor de δ se ha reducido al valor $\delta = 0.0025$.

Como puede apreciarse, se ha reducido la amplitud de las oscilaciones, pero a costa de un comportamiento excesivamente lento del sistema controlado, como se observa en la figura 8.

Un análisis de los resultados anteriores lleva a la conclusión de que sería conveniente que el valor de δ no sea constante sino que decrezca a medida que el error de regulación se acerque a cero. Para lograr ese tipo de comportamiento de δ , es posible utilizar la función:

$$\delta(t) = \delta_1 e(t)^2 \quad (24)$$

donde $e(t) = r(t) - y(t)$, es el error de regulación y δ_1 es constante.

El comportamiento mejorado resultante de este cambio se observa en la figura 9, al que solamente hay que objetar las sobrecrestas que aparecen en las transiciones. Una forma simple e intuitiva de mejorar aun más esta respuesta consiste en modificar las restricciones sobre las variables de control y controlada. En este caso, después de algunos intentos, se obtuvo la respuesta prácticamente perfecta que aparece en la figura 11, utilizando los límites:

$$\begin{aligned} u_{\max} &= 0.82 \\ u_{\min} &= 0.35 \\ y_{\max} &= 11 \\ y_{\min} &= 0 \end{aligned}$$

La imposición de restricciones más fuertes en u y y , puede conducir en algunos casos a la necesidad de incrementar el tamaño de la población utilizada, pues es presumible que una parte de los individuos sean excluidos por violar alguna de las restricciones; esto significa que el tamaño efectivo

de la población puede reducirse. En este ejemplo particular, sin embargo, fue posible mantener invariable el tamaño de la población en 10, lo que resulta muy conveniente.

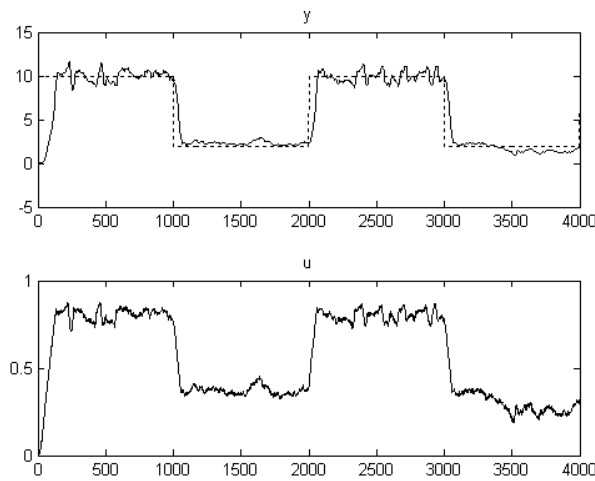


Figura 7. Repuesta del sistema con retardo de 10 unidades y $\delta=0.01$

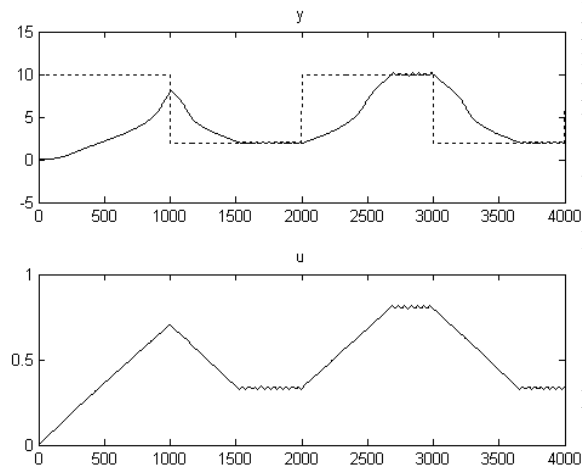


Figura 8. Repuesta del sistema con retardo de 10 unidades y $\delta=0.001$

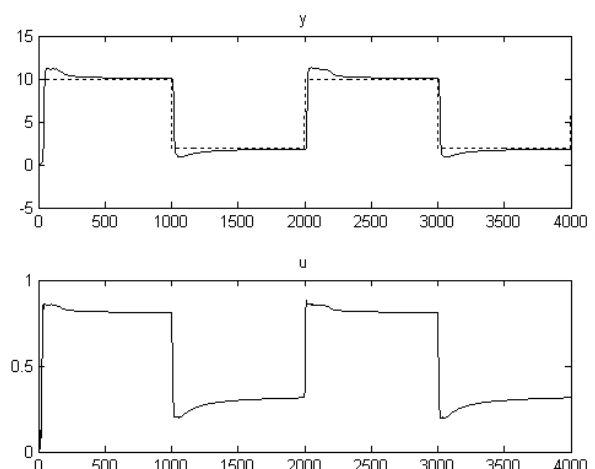


Figura 9. Repuesta del sistema con retardo de 10 unidades, espacio de búsqueda variable según ecuación (24) y $\delta_1=0.001$

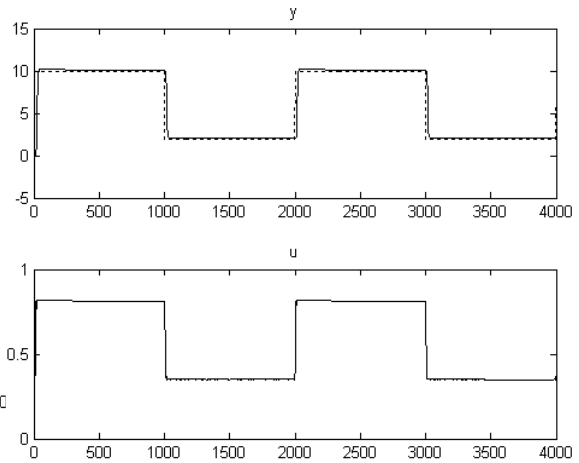


Figura 10. Respuesta del sistema con retardo de 10 unidades, espacio de búsqueda variable según ecuación (24), $\delta_1=0.001$, $u_{max}=0.82$, $u_{min}=0.35$, $y_{max}=11$, $y_{min}=0$

6. CONTROL NG DE UNA UNIDAD CALDERA - TURBINA

El ejemplo mostrado anteriormente ha servido para presentar la metodología de control predictivo neuro-genético que se propone en este trabajo, pero no resulta quizás suficientemente ilustrativo, ya que se trata del control de un sistema no lineal discreto creado ad-hoc. A continuación vamos a aplicar el método a un sistema real y continuo, descrito en el espacio de estados, que hará posible hacer un análisis comparativo con otros métodos de control muy bien establecidos en la práctica como son los controladores PI y GPC: se trata del control de una unidad caldera con domo – turbina cuyo esquema simplificado se presenta en la figura 11.

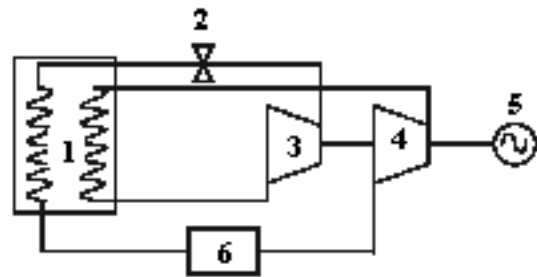


Figura 11. Unidad caldera turbina

En el esquema de la figura 11, el vapor se genera en la caldera 1, pasa a través de la válvula 2 y se expande en la turbina de alta presión 3. Después del recalentamiento en la caldera 1, una expansión adicional tiene lugar en las turbinas de presión baja e intermedia representadas en 4, seguida por la condensación en 6. La electricidad es generada en el alternador 5.

Para el propósito de este trabajo, vamos a utilizar el modelo de segundo orden, no lineal, presentado

en el trabajo de Wigren (2005), que se ofrece a continuación:

$$\begin{aligned}\dot{x}_1(t) &= 0.014u_1(t) - 0.0033(x_1(t))^{1.125}u_2(t) \\ \dot{x}_2(t) &= 0.01408(x_1(t))^{1.125}u_2(t) - 0.064x_2(t) \\ y(t) &= 0.2957(x_1(t))^{1.125}u_2(t) + 3.456x_2(t) + v(t)\end{aligned}\quad (25)$$

En este modelo, $x_1(t)$ es la presión en el domo en kg/m^2 , $x_2(t)$ es la presión en el precalentador, $u_1(t)$ es el flujo de fuel-oil en tons/hora (no representado en el esquema), $u_2(t)$ es la posición de la válvula $\in [0,1]$ y $y(t)$ es la potencia de salida en MW. En el estudio de simulación, la válvula $u_2(t)$ se fijó en el valor 0.8 y el control de la potencia generada se realiza variando el flujo de fuel-oil. Se incluye en el modelo, además, una perturbación $v(t)$ que representa a las variaciones en el consumo y que se modela mediante un proceso aleatorio de incrementos independientes (drift):

$$v(t) = v(t-1) + e(t) \quad (26)$$

donde $e(t)$ es ruido blanco discreto con varianza igual a 0.2.

En la figura 12 se observa la respuesta dinámica de la potencia generada, ante cambios en el flujo de fuel-oil entre 3.6 y 4.6 tons/hora. Como puede observarse, la respuesta se estabiliza después de transcurridos unos 2000 seg aproximadamente. Teniendo en cuenta esta información, el período de control para este proceso se fijó en 20 segs.

En la figura 13 se presenta el resultado del entrenamiento de un modelo neuronal para la unidad caldera-turbina, a partir de una muestra de 2000 datos tomados con un período de 20 segs. y utilizando la metodología explicada en el epígrafe 2. La señal de entrada utilizada es un ruido uniformemente distribuido entre 0.1 y 0.3, superpuesto al valor base 4. Los parámetros utilizados para el modelo neuronal fueron:

Orden del modelo neuronal: $n = 2$
 Neuronas en la capa de entrada: $N_{ent} = 5$
 Neuronas en la capa escondida: $N_{escon} = 4$
 Tamaño de la muestra de datos: $N_{muestr} = 2000$
 Datos para entrenamiento: $N_{entren} = 1600$
 Datos para validación: $N_{val} = 400$
 Coeficiente de aprendizaje inicial = 0.2
 Función de activación de las neuronas de la capa escondida: sigmoide
 Función de activación de la neurona de salida: lineal

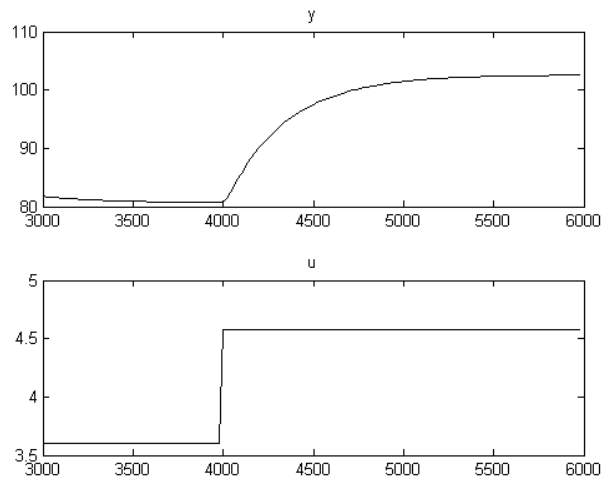


Figura 12. Respuesta de la potencia generada ante cambios en escalón en el flujo de fuel-oil

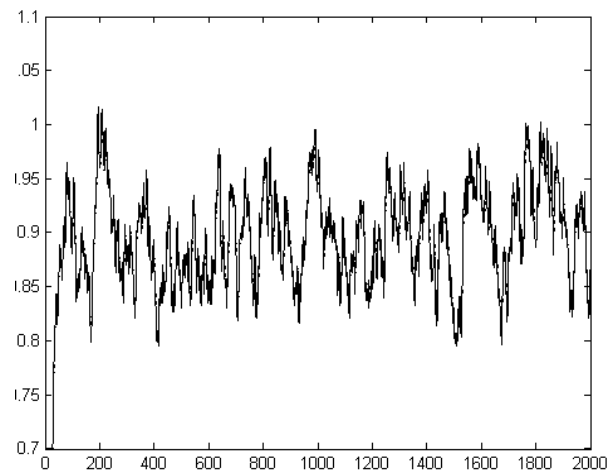


Figura 13. Respuestas del proceso y del modelo neuronal superpuestas

El error medio cuadrático obtenido después de aproximadamente 2000 épocas de entrenamiento fue de $6.2 \text{ E-}5$.

Como puede apreciarse en la figura 13 existe una coincidencia prácticamente perfecta entre las salidas del proceso y del modelo neuronal.

A continuación vamos a hacer un estudio comparativo de simulación del comportamiento del controlador NG con los controladores PI continuo y GPC, considerando dos escenarios: cambios en la referencia entre 80 y 100 Mw y referencia constante más una perturbación en la carga que varía en forma aleatoria de acuerdo con el modelo dado en la ecuación (26). Una realización típica de la perturbación en la carga se muestra en la figura 14.

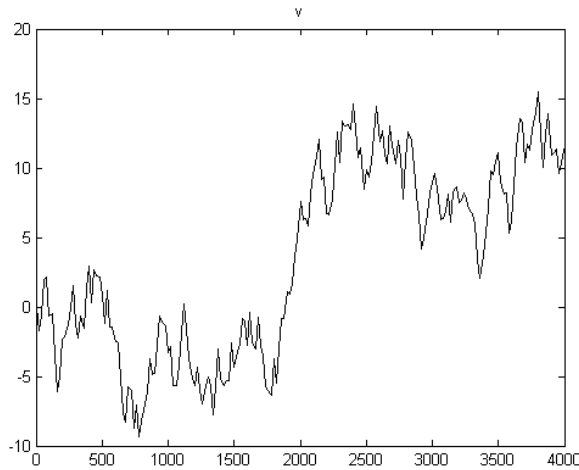


Figura 14. Perturbación típica en la carga

En el control de este proceso se consideran además las siguientes restricciones: El flujo de fuel-oil no puede sobrepasar las 7 tons/hora y la potencia generada no debe exceder a 110 Mw. Es decir:

$$\begin{aligned} 0 \leq u_1(t) \leq 7 \\ 0 \leq y(t) \leq 110 \end{aligned} \quad (27)$$

El período de control para el GPC y el NG fué de 20 segundos.

6.1 Cambios en la referencia entre 80 y 100 Mw.

En la figura 15, se observa el comportamiento de las variables controlada y de control cuando se utiliza un regulador PI continuo, con parámetros $K_p = 0.5$ y $T_i = 320$. Se utilizó también un limitador de la variable de control teniendo en cuenta las restricciones dadas en (27). Valores mayores de la ganancia o menores del tiempo integral no mejoran apreciablemente el comportamiento de la respuesta. El error medio cuadrático en los últimos 3000 segs. fue de aproximadamente 29.

En la figura 16 se presenta el comportamiento de un controlador GPC, que usa como predictor un modelo lineal ARX con orden $n = n_A = n_B = 3$. Los horizontes de predicción y control se seleccionaron como $N_1 = 1$, $N_2 = 12$ y $N_u = 3$. El coeficiente de peso de los incrementos del control se fijó en $w = 800$. Este valor excesivamente grande de w se hizo necesario para mantener la estabilidad del sistema: valores ligeramente inferiores dieron como resultado un comportamiento oscilatorio o incluso inestable. Tampoco resultó útil aumentar los valores de N_2 o N_u . Como puede observarse, el comportamiento del GPC en este caso es excesivamente lento, peor que el del controlador PI.

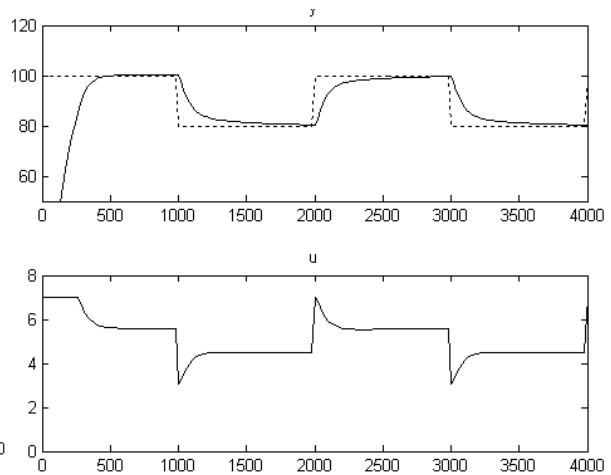


Figura 15. Control PI de la potencia generada con parámetros: $K_p = 0.5$, $T_i = 320$, $e_m^2 = 29$

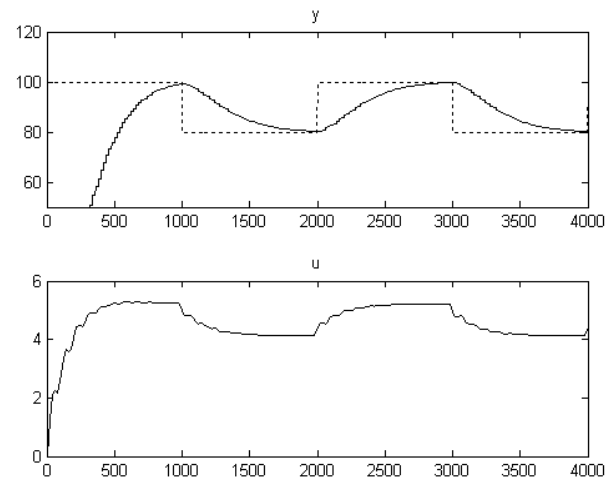


Figura 16. Control GPC de la potencia generada con parámetros: $N_1 = 1$, $N_2 = 12$, $N_u = 3$, $e_m^2 = 89$

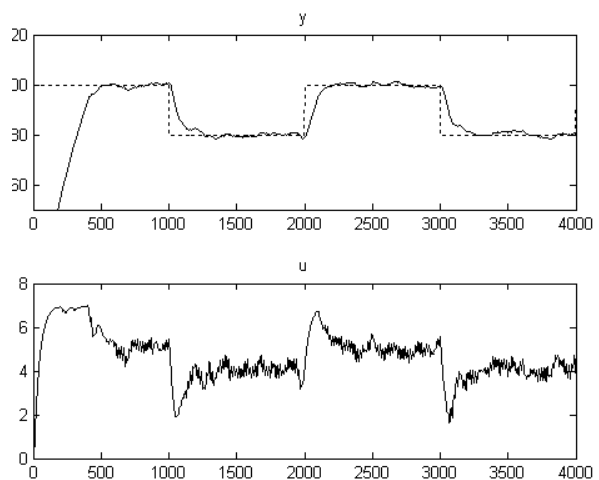


Figura 17. Control NG con parámetros: $N_{ind} = 50$, $N_{ogen} = 5$, $\delta = 0.2$, $N_2 = 12$, $N_u = 3$, $e_m^2 = 17.1$

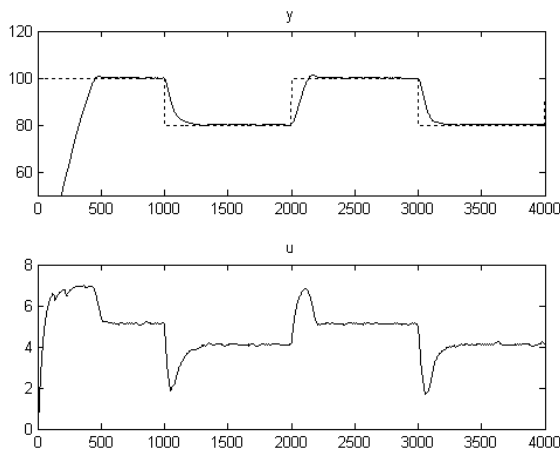


Figura 18. Control NG. $N_{ind}=50$, $N_{ogen}=5$, $\delta=0.2$, $N_2=12$, $N_u=3$, $cf=0.1$, $dr=10$, $e_m^2=14.2$

En la figura 17 aparece el comportamiento típico de un controlador NG cuyos horizontes de tiempo son iguales a los del GPC y los parámetros del algoritmo genético son: $N_{ind} = 50$, $N_{ogen} = 5$, $\delta = 0.2$. El criterio a minimizar es el cuadrático expresado en la ecuación (12) siendo el error medio cuadrático de esta realización $e_m^2=17.1$.

Aunque se ha logrado una disminución del error medio cuadrático, la respuesta presenta oscilaciones alrededor de la referencia que son más evidentes en el control y que resultan indeseables en la práctica.

Resulta claro que una disminución del rango de los incrementos del control en los alrededores de la referencia contribuiría a disminuir la amplitud de las oscilaciones. En el epígrafe 5 se introdujo ventajosamente la sustitución de un valor constante de δ , por la función cuadrática dada en (24). Esta solución, sin embargo, no ofreció los resultados deseados en este ejemplo.

Otra posibilidad de disminuir el rango de los incrementos cuando el error de regulación es pequeño, consiste en utilizar la siguiente regla heurística:

$$\text{Si } \text{abs}(r(t) - y(t)) < cf, \text{ entonces } \delta = \delta/dr \tag{27}$$

Los valores de cf y dr pueden determinarse experimentalmente en cada aplicación. Por ejemplo en el caso de la caldera, partiendo de un valor mayor de δ , por ejemplo 0.3, es posible lograr un despegue más rápido y cuando el error es de 10 unidades ($cf = 10$), la disminución de δ en un factor de 10 ($dr = 10$) frena la respuesta y ésta “atteriza” suavemente en el valor deseado, prácticamente sin oscilaciones. La figura 18 ilustra el mejoramiento obtenido mediante la

introducción de la regla heurística (27), lográndose una disminución adicional del e_m^2 y una atenuación apreciable en las oscilaciones del control.

La realización de 20 repeticiones de la simulación de este método, con los mismos parámetros, dió como resultado un valor promedio de $e_m^2=14.15$

6.2 Control a referencia constante y perturbaciones en la carga

En las figuras 19, 20 y 21 se muestran realizaciones típicas de los tres controladores considerados, cuando la referencia se mantiene constante y el control tiene que lidiar con perturbaciones aleatorias en la carga tales como la mostrada en la figura 14.

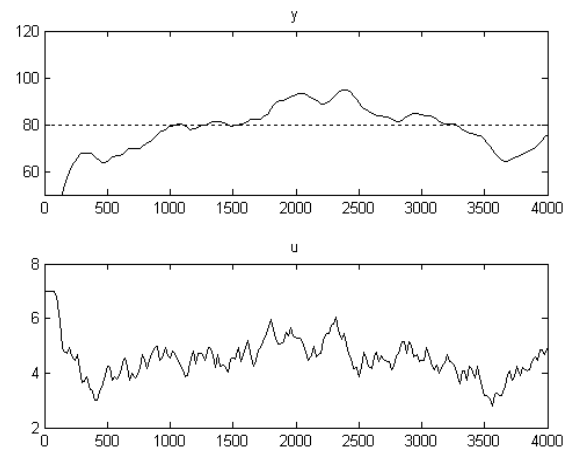


Figura 19. Control PI de la potencia generada con parámetros: $K_p = 0.5$, $T_i = 320$, perturbación en la carga, $e_m^2=14.56$

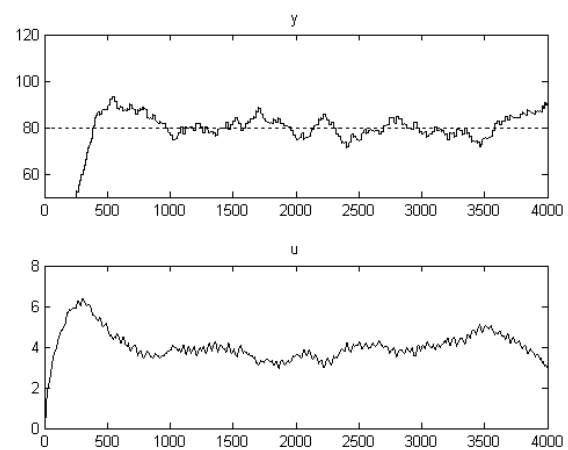


Figura 20. Control GPC de la potencia generada con parámetros: $N_1=1$, $N_2=12$, $N_u=3$, $e_m^2=16$

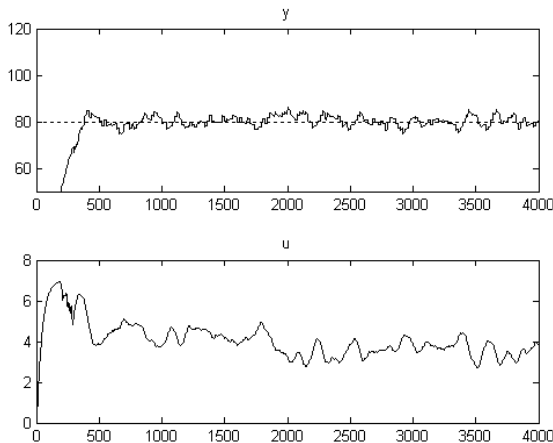


Figura 21. Control NG de la potencia generada con parámetros: $N_{ind}=50$, $N_{ogen}=5$, $\delta\tau=0.2$, $N_2=12$, $N_u=3$, $e_m^2=5.1$

El comportamiento promedio de los tres reguladores después de 20 simulaciones, se ofrece en la Tabla 2.

Tabla2. Comparación de reguladores

Regulador	e_m^2
PI	27.7
GPC	86.3
NG	4.98

7. CONCLUSIONES

El objetivo fundamental del presente trabajo ha sido demostrar la viabilidad de un enfoque neuro-genético para la solución del problema del control predictivo, en aquellos casos en que otras soluciones convencionales no resultan satisfactorias.

Aunque la investigación sobre este tema es todavía incipiente y no es posible, por tanto, dar conclusiones definitivas, los resultados mostrados en el presente trabajo permiten, al menos, hacer las siguientes afirmaciones:

- Una solución al control predictivo basada en el entrenamiento de un modelo neuronal como modelo predictor y el uso de un algoritmo genético para el cálculo del control óptimo, puede ser realizada utilizando recursos de cómputo que están disponibles en la actualidad. En efecto, se han mostrado ejemplos en el que incluso con poblaciones muy reducidas, el algoritmo genético utilizado es capaz de llegar a la solución óptima, sin un costo de cómputo excesivo.
- El uso de una población de incrementos del control, una de las novedades que proponemos en este trabajo, facilita en gran medida la

posibilidad de aplicar esta metodología en tiempo real, al permitir la reducción del tamaño de la población y hacer depender en buena medida la dinámica del lazo cerrado del tamaño del espacio de búsqueda.

- Gracias a la versatilidad de los algoritmos genéticos es posible mejorar el comportamiento del regulador NG acudiendo a soluciones heurísticas ad-hoc tales como modificar el criterio de optimización o el espacio de búsqueda, restringir adecuadamente las variables de control y controlada, etc. Algunas de esas soluciones han sido exploradas exitosamente en este trabajo, pero resulta indudable que queda todavía mucho espacio para la imaginación y la experimentación en el sentido de mejorar aun más las cualidades de esta metodología de control.
- La mayor versatilidad y generalidad del control NG, se paga sin duda con la necesidad de definir un mayor número de parámetros, lo que resulta en una complejidad adicional que no será posible justificar salvo en aquellos casos en que pueda obtenerse una mejora sensible en los resultados.
- El costo computacional del NG, a pesar de las simplificaciones, sigue siendo alto en comparación con otros métodos. Para que se tenga una idea aproximada, el tiempo requerido para realizar el control de la caldera, con una población de 50 individuos y 5 generaciones en una computadora estándar, es de 125 segs. para los 4000 períodos de tiempo simulados, lo que es aproximadamente 8 veces mayor que el tiempo requerido para la simulación del GPC. No obstante, en el caso de procesos industriales típicos, como el ejemplo de la caldera, la cuestión del tiempo requerido no es una limitante con la tecnología de cómputo disponible.
- Dados los resultados obtenidos en las simulaciones, tenemos una confianza razonable en que el enfoque presentado puede ser una alternativa práctica en aquellos casos en que otras soluciones convencionales fallan. Se requiere, no obstante, más estudio experimental que incluya la aplicación de este esquema en condiciones reales de control.

REFERENCIAS

- Aguado, A., Hernández, V., Ordaz, A. y Noriega, A. (1999): Control Predictivo Neuro-genético, *Memorias del II Simposio de Control Automático*, CIMAF'99, La Habana.
- Aguado, A. (2000) Temas de Identificación y Control Adaptable. Editorial ICIMAF, La Habana.

- Blasco F.X., (1999): Control predictivo basado en modelos mediante técnicas de optimización heurística. PhD tesis. Universidad Politécnica de Valencia.
- Back T., (1996): *Evolutionary Algorithms in Theory and Practice*, Oxford University Press, New York,
- Baker E., (1987): Reducing Bias in the Selection Algorithms, *Proceedings of the Second International Conference on Genetic Algorithms*, pp. 14-21, Vol. 11, Grefenstette.
- Camacho, E. F. y Bordons, C. (2004): Control Predictivo, pasado, presente y futuro, Copyright CEA-IFAC.
- Clarke, D.W., Mohtadi C., Tuffs, P.S., (1987a): *Generalized Predictive Control-part I*. Automatica 23 (2), 137-148.
- Clarke, D.W., Mohtadi, C., Tuffs,P.S., (1987b): *Generalized Predictive Control-part II*, Extensions and Interpretations. Automatica 23 (2), 137-148.
- Martínez, M., Senent, J.S y Blasco, X. (1998): *Predictive controller based on a genetic algorithm*, Engineering Applications of Artificial Intelligence 11, 355-367.
- Martínez, M., Senent, J.S. y Blasco, X. (2006): *Algoritmos genéticos aplicados al diseño de controladores robustos*, Revista RIAI, 3(1), 39-51.
- Shin, S.C. and Park, S.B.(1998): *GA - based predictive control for nonlinear processes*, Electronics Letters, Vol. 34 No. 20
- Wigren, T. (2005): *Recursive identification based on nonlinear state space models applied to drum-boiler dynamics with nonlinear output equations*, Proceedings of the American Control Conference, Portland, OR.
- Zamarreño, J. M. and Vega P., 1999): *Neural predictive control. Application to a highly non-linear system*, Engineering Applications of Artificial Intelligence, Volume 12, Issue 2, Pages 149-158.