

# Control adaptable utilizando Redes Neuronales Artificiales Polinomiales

E. Gómez-Ramírez

Universidad La Salle, Jefe del Área de Ingeniería y Tecnología  
<egomez@aldebaran.ci.ulsal.mx>

A. S. Poznyak

CINVESTAV-IPN, Sección de Control Automático,  
Av. IPN 2508 AP 14-740, CP 07000, México D.F., México.  
<apoznyak@ctrl.cinvestav.mx>

R. Lozano

Université de Compiègne  
Centre de Recherches de Royallieu BP 20529, 60205 Compiègne Cedex France.  
<Rogelio.Lozano@hds.utc.fr>

## RESUMEN

Existen en la literatura de Control Adaptable, diferentes procedimientos en los que es posible identificar un sistema lineal. El problema fundamental es que una cantidad importante de fenómenos de la vida real son de tipo no lineal y no es tan sencillo el modelar este tipo de dinámicas.

En este trabajo se presenta una forma de identificar sistemas no lineales utilizando las propiedades de las Redes Neuronales Artificiales y las técnicas de Algoritmo Genético en la optimización de su arquitectura. Adicionalmente se presenta una técnica novedosa de control adaptable para cancelar la dinámica no lineal del sistema y colocar los polos en el punto de operación deseado. Se presenta el comportamiento del algoritmo para el caso lineal y no lineal y finalmente se analiza la importancia teórica y operacional de estas técnicas.

*Palabras clave:* Sistemas no lineales, Redes Neuronales Artificiales, Algoritmo Genético, Arquitectura, Control Adaptable.

## ABSTRACT

In Adaptive Control Theory exists different procedures to identify a linear system. The fundamental problem is that in the real world many systems are not linear and it is not easy to obtain the mathematical model.

In this work, an identification procedure for a non linear system is presented using the properties of Artificial Neural Networks and Genetic Algorithm to optimize the architecture of the network. A new technique of Adaptive Control to cancel the non linear dynamics of the system is proposed to set the poles of the system in a desire position. The behavior of the algorithm for the linear and non linear case is presented with the analysis of the theoretical and operational importance of these techniques.

*Keywords:* Non linear system, Artificial Neural Networks, Genetic Algorithm, Architecture, Adaptive Control.

## INTRODUCCIÓN

Para aplicar un controlador es muy importante conocer la dinámica del sistema. Por esta

razón es necesario identificar la mejor forma posible el modelo matemático de la planta o al menos conocer la relación entre variables internas, salidas y entradas. Actualmente existen

alternativas del área de Control Inteligente (1) que intentan sustituir este paso por el conocimiento de algún experto y modelar ciertas relaciones de la dinámica. Específicamente la lógica difusa (2) por medio de reglas mapea este conocimiento para encontrar cuál es la relación de control para cada caso. También existen otras herramientas como las Redes Neuronales Artificiales (3) que pueden aprender la dinámica de la planta e inclusive la de un controlador que modifique la respuesta de la planta a ciertas características deseadas.

Estas técnicas de aprendizaje de la dinámica de un sistema han competido y compartido campos y algoritmos con el área de control adaptable (4). Esta área ha tenido gran auge por ser una aproximación sistemática para el ajuste automático de los parámetros del controlador en tiempo real. Este ajuste puede depender del conocimiento de la dinámica, de un modelo identificado de la planta, o simplemente puede adaptarse a los cambios del sistema dependiendo de su interacción con el medio.

En este artículo se presenta una combinación de Redes Neuronales Artificiales Polinomiales con técnicas de control adaptable y su uso para sistemas lineales y no lineales.

## CONTROL ADAPTABLE

Las incertidumbres y variaciones de los parámetros de procesos hacen que el desempeño de los sistemas de control se reduzca. Una de las alternativas para reducir estas perturbaciones y oscilaciones sobre las variables de control es la retroalimentación. De tal forma que si es posible encontrar el valor de estas perturbaciones, *alimentan* al controlador para ser canceladas y obtener un apropiado comportamiento de nuestro control. Un concepto necesario para cuantificar el buen funcionamiento de un sistema es el Índice de Desempeño (ID). Este indicador es una medición del comportamiento del sistema; se compara con el ID deseado y la diferencia es introducida al mecanismo de adaptación. Este mecanismo actuará sobre los parámetros del controlador para modificar el desempeño del sistema.

## Definición 1

Un sistema de control adaptable mide cierto ID del sistema de control utilizando las entradas, los estados, las salidas y las perturbaciones conocidas. De la comparación del ID medido y del deseado, el mecanismo de adaptación modifica los parámetros ajustables del controlador y/o genera un control auxiliar para mantener el ID del sistema de control lo más cercano posible al deseado.

Un Sistema de control adaptable puede ser visto como un sistema jerárquico de dos niveles donde:

- El Nivel 1 es un Control Retroalimentado Directo
- El Nivel 2 es el Lazo de Adaptación

En la práctica frecuentemente se tiene un tercer nivel de supervisión, el cual decide si las condiciones cumplen con las características de operación del lazo de adaptación.

Existen varios esquemas utilizados en Control Adaptable como:

- Control Adaptable de Lazo Abierto
- Control Adaptable Directo
- Control Adaptable Indirecto

## Control Adaptable de Lazo Abierto

Este Mecanismo de control es una simple tabla de relaciones entrada-salida almacenada en la computadora, la cual define los parámetros de control para un conjunto de mediciones del medio. Esta técnica considera la existencia de una relación definida entre algunas variables que caracterizan el medio y los parámetros del modelo de la planta. Utilizando esta relación es posible reducir los efectos de la variación de los parámetros sobre el desempeño del sistema, modificándolos dependiendo de las condiciones.

### Control Adaptable Directo

En este control los parámetros dependen directamente de la especificación del desempeño deseado del lazo de control. Este desempeño puede depender de un modelo de referencia con las características dinámicas deseadas. De tal forma que el diseño del controlador debe cumplir que:

- El error entre la salida de la planta y la salida del modelo de referencia sea igual a cero para condiciones iniciales idénticas
- Cualquier error inicial desaparezca con cierta dinámica.

Cuando no se conocen los parámetros de la planta o éstos son variantes en tiempo, y se desea mantener el ID deseado, es necesario utilizar un esquema de control adaptable conocido como Control Adaptable con Modelo de Referencia (Model Reference Adaptive Control, MRAC). Este esquema se basa en la observación de que la diferencia entre la salida de la planta y la salida del modelo de referencia es una medición entre el desempeño real y el deseado. Esta información es utilizada por el mecanismo de adaptación para directamente ajustar los parámetros del controlador en tiempo real para forzar asintóticamente el error a cero. Este modelo es el prototipo básico del Control adaptable directo. En la Figura 1 se puede observar el diagrama para un sistema lineal aplicando un control adaptable directo.

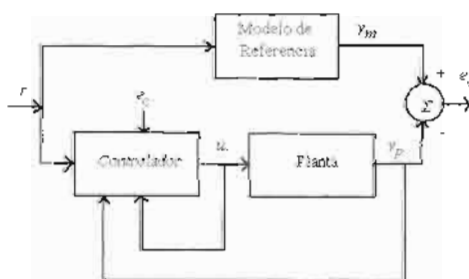


Figura 1 Control Adaptable directo

Los esquemas de Control Adaptable Directo se obtienen principalmente de las siguientes dos formas:

- Definiendo una ecuación para una señal de error (error de adaptación), la cual es una función de la diferencia entre los parámetros sintonizados del controlador y los parámetros actuales
- Utilizando una aproximación de un control adaptable indirecto con un predictor adaptable de la salida de la planta reparametrizada, en términos de los parámetros del controlador, y forzando la salida del predictor adaptable para seguir exactamente la trayectoria deseada.

### Control Adaptable Indirecto

Este esquema es denominado indirecto debido a que utiliza los siguientes pasos:

1. Estimación en línea de los parámetros de la planta y posteriormente,
2. Cálculo en línea de los parámetros del controlador utilizando la estimación del modelo de la planta.

Este sistema de control ofrece una gran variedad de combinaciones de leyes de control y de técnicas de estimación de parámetros, debido a que es posible escoger en principio cualquier sistema de estimación de parámetros con cualquier estrategia de control. En la Figura 2 se puede observar el diagrama correspondiente.

El problema esencial de un esquema de control adaptable es el asegurar la estabilidad del sistema de lazo cerrado. Para esto sería necesario un profundo y complejo análisis del controlador. Esto se puede reducir si se cumplen ciertas propiedades.

Uno de los esquemas de control adaptable indirecto más utilizados es el de Colocación adaptable de polos. En las siguientes secciones se describirá esta metodología para:

- Sistemas lineales de fase mínima.
- Sistemas lineales de fase no mínima.

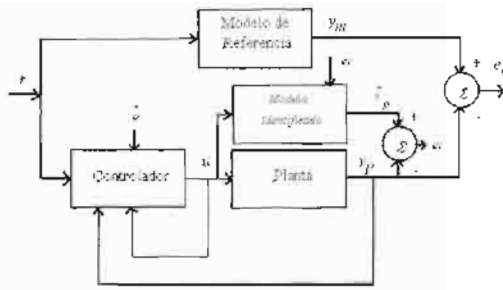


Figura 2 Control Adaptable Indirecto

Control de Sistemas Lineales de Fase Mínima en Tiempo Discreto.

Considérese el siguiente sistema como una entrada una salida (single input single output, SISO):

$$A(q^{-1})y_k = q^{-d}B(q^{-1})x_k \tag{Ec. 1}$$

donde:  $x_k$  es la entrada,  $y_k$  es la salida,  $q^{-1}$  es el operador de retardo,  $d$  es el retardo y

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{n_2}q^{-n_2} \\ B(q^{-1}) &= b_0 + b_1q^{-1} + \dots + b_{n_1}q^{-n_1} \end{aligned} \tag{Ec. 2}$$

donde  $n_2$  es el número de polos y  $n_1$  el número de ceros del sistema.  $B(q-1)$  es estable con  $b_0 \neq 0$ .

Considérese también que se cumple la siguiente relación:

$$C(q^{-1}) = A(q^{-1})S(q^{-1}) + q^{-d}R(q^{-1}) \tag{Ec. 3}$$

donde:

$$\begin{aligned} C(q^{-1}) &= 1 + c_1q^{-1} + \dots + c_{n_2+d-1}q^{-(n_2+d-1)} \\ S(q^{-1}) &= s_0 + s_1q^{-1} + \dots + s_{d-1}q^{-(d-1)} \\ R(q^{-1}) &= r_0 + r_1q^{-1} + \dots + r_{n_1-1}q^{-(n_1-1)} \end{aligned} \tag{Ec. 4}$$

.Con esta identidad polinomial se pueden calcular los valores de  $S(q^{-1})$  y  $R(q^{-1})$  para un valor determinado de  $C(q^{-1})$ .

Si se calcula  $C(q^{-1})y_{i-d}$

$$\begin{aligned} C(q^{-1})y &= A(q^{-1})S(q^{-1})y + q^{-d}R(q^{-1})y \\ C(q^{-1})y &= S(q^{-1})A(q^{-1})y + q^{-d}R(q^{-1})y \\ C(q^{-1})y &= S(q^{-1})[q^{-d}B(q^{-1})x] + q^{-d}R(q^{-1})y \\ C(q^{-1})y &= q^{-d}[S(q^{-1})B(q^{-1})x + R(q^{-1})y] = q^{-d}y^* \end{aligned} \tag{Ec. 5}$$

$$y^* = S(q^{-1})B(q^{-1})x + Ry \tag{Ec. 6}$$

y si  $C(q^{-1})=1$  entonces

$$y_{k+d} = B(q^{-1})S(q^{-1})x_k + R(q^{-1})y_k \tag{Ec. 7}$$

Esta ecuación describe el comportamiento de la salida del sistema en el tiempo  $k+d$ .



Figura 3 Esquema de Control Adaptable para Sistemas de Fase Mínima

Control de Sistemas Lineales de Fase no Mínima en Tiempo Discreto.

Considérese de igual forma que en el caso de fase mínima el siguiente sistema una entrada una salida:

$$A(q^{-1})y_k = B(q^{-1})x_k \quad (\text{Ec. 8})$$

donde:  $x_k$  es la entrada,  $y_k$  es la salida,  $q^{-1}$  es el operador de retardo,  $d$  es el retardo y

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{n_2}q^{-n_2}$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + \dots + b_{n_1}q^{-n_1} \quad (\text{Ec. 9})$$

donde  $n_2$  es el número de polos y  $n_1$  el número de ceros del sistema.  $B(q^{-1})$  es estable con  $b_0 \neq 0$ .

Considérese también que se cumple la siguiente relación:

$$C(q^{-1}) = A(q^{-1})S(q^{-1}) + B(q^{-1})R(q^{-1}) \quad (\text{Ec. 10})$$

donde:

$$S(q^{-1}) = s_0 + s_1q^{-1} + \dots + s_{d-1}q^{-(n_1-1)}$$

$$R(q^{-1}) = r_0 + r_1q^{-1} + \dots + r_{n_2-1}q^{-(n_2-1)} \quad (\text{Ec. 11})$$

donde:  $n_3 = \max(n_1, n_2)$

Con esta identidad polinomial se pueden calcular los valores de  $S(q^{-1})$  y  $R(q^{-1})$  para un valor determinado de  $C(q^{-1})$  donde  $A(q^{-1})$  y  $B(q^{-1})$  son primos en  $re$  si.

Si se calcula  $C(q^{-1})y_{1+d}$

$$C(q^{-1})y = A(q^{-1})S(q^{-1})y + B(q^{-1})R(q^{-1})y$$

$$C(q^{-1})y = S(q^{-1})[B(q^{-1})x] + B(q^{-1})R(q^{-1})y$$

$$C(q^{-1})y = B(q^{-1})[S(q^{-1})x + R(q^{-1})y] \quad (\text{Ec. 12})$$

Si se propone la siguiente ley de control:

$$r = S(q^{-1})x + R(q^{-1})y \quad (\text{Ec. 13})$$

se tiene que:

$$C(q^{-1})y = B(q^{-1})r \quad (\text{Ec. 14})$$

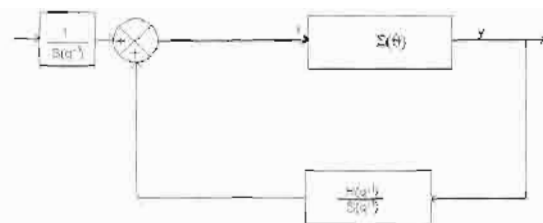


Figura 4 Esquema de Control Adaptable para Sistemas de Fase no Mínima

#### Control Adaptable con Redes Neuronales Artificiales

Como se mencionó anteriormente, en el control adaptable indirecto es posible utilizar cualquier algoritmo que cumpla con ciertas condiciones para la estimación del modelo de la planta. Las teorías y técnicas conocidas como RNA han demostrado su gran capacidad para este tipo de tareas. La característica más significativa de las RNA es su habilidad para aproximar cualquier función no lineal. Esta habilidad las ha colocado como una herramienta muy útil para el modelado de sistemas no lineales, la cual es muy importante en el diseño de controladores para este tipo de dinámicas (5)(6). Funahashi, Hornik y cols., Cybenko, Cotter (7-10) y Blum y Li, utilizando el Teorema de Weierstrass como base, demostraron que una función continua puede ser aproximada utilizando una red neuronal estática con una capa oculta. Otros autores han utilizado el teorema de Kolmogorov (11) para demostrar las capacidades de estas redes. Modelos estáticos y dinámicos han sido propuestos para identificar y controlar sistemas dinámicos con diferentes arquitecturas.

Un esquema muy utilizado para la identificación de una función no lineal es el que se presenta en la Figura 5.

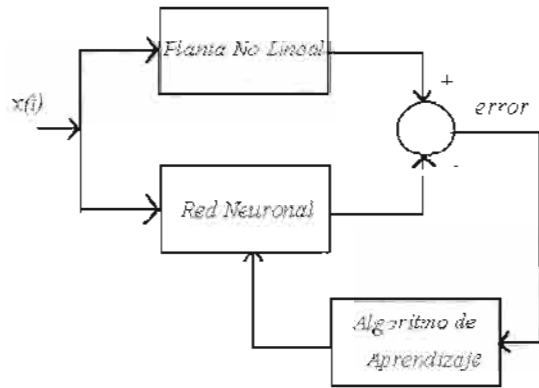


Figura 5 Esquema General de Aprendizaje para la Aproximación de una Función Utilizando Redes Neuronales

De forma similar a las secciones previas es posible desarrollar algoritmos de Control Adaptable con RNA. En la Figura 6 y Figura 7 se presentan dos esquemas de control adaptable directo e indirecto utilizando RNA (12)

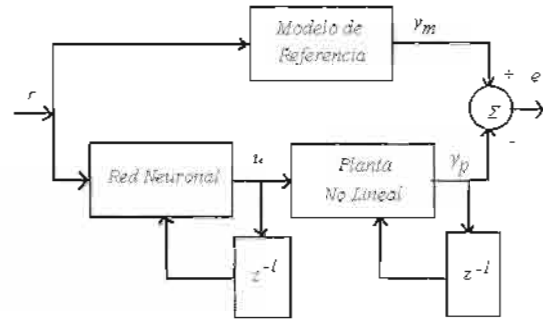


Figura 6 Esquema de Control Adaptable Directo de una Planta no Lineal Utilizando RNA

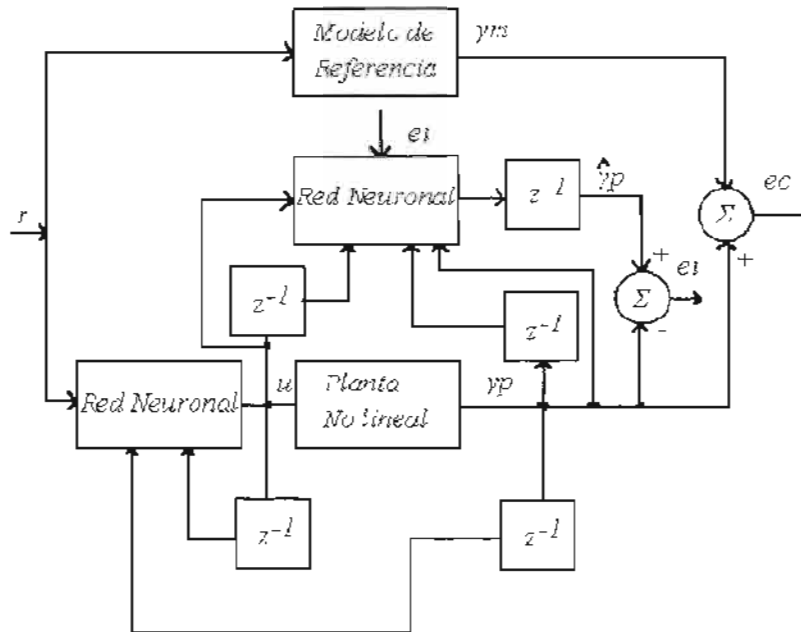


Figura 7 Control Adaptable Indirecto Utilizando RNA

Como se puede observar la estructura es muy similar a la utilizada para el caso lineal y en lugar de las ganancias lineales se coloca una red neuronal artificial.

Los bloques con  $z^{-1}$  representan los retardos de la señal. Uno de los algoritmos más utilizados para el entrenamiento de estas redes es el *backpropagation*. Como se puede ver en las figuras los datos de entrenamiento entrada-salida son generados directamente del sistema dinámico. En (12) se puede consultar la forma en que pueden ser aplicados para distintas plantas no lineales

#### Modelos NARMAX: Teoría

Algunos autores han considerado a los modelos NARMAX (13)(14) como una alternativa eficiente para la identificación de sistemas dinámicos no lineales (15)(16). En esta sección se describe lo que son estos modelos por la importancia que tienen en el área de identificación y por su similitud con las redes neuronales (17), en especial con las RNAP.

Considerando algunas suposiciones un sistema de control no-lineal discreto estocástico puede ser descrito mediante el modelo NARMAX (18):

$$y(t) = f \left[ \begin{array}{l} y(t-1), \dots, y(t-n_y), u(t-1), \dots, \\ u(t-n_u), e(t-1), \dots, e(t-n_e) \end{array} \right] + e(t)$$

(Ec. 15)

donde  $y(t)$ ,  $u(t)$  y  $e(t)$  son la salida, entrada y ruido del sistema respectivamente;  $n_y$ ,  $n_u$ ,  $n_e$  son los órdenes o valores anteriores máximos de la salida, entrada y ruido. En este caso  $e(t)$  se supone que sea una secuencia blanca y  $f()$  es alguna función no lineal. Este modelo se le denomina NARMAX por su semejanza con el modelo lineal:

$$y(k) = a_0 + \sum_{i=1}^{n_y} a_i y(k-i) + \sum_{i=1}^{n_u} b_i u(k-i)$$

(Ec. 16)

Descomponiendo  $f()$  como un polinomio de grado  $L$  se obtiene la siguiente representación:

$$y(t) = \sum_{i=1}^{n_0} \theta_i x_i(t) + \varepsilon(t, \theta)$$

(Ec. 17)

donde:

$$\begin{aligned} n_0 &= \sum_{i=0}^L n_i; \quad n_0 = 1, \\ n_i &= n_{i-1} (n_y + n_u + n_e + i - 1) / i \\ i &= 1, \dots, L \end{aligned}$$

(Ec. 18)

$$\theta = (\theta_1, \dots, \theta_{n_0})^T$$

(Ec. 19)

Como se observa puede haber varias representaciones para un mismo sistema. Existen algunos casos particulares para representar  $f(*)$  utilizando el modelo bilineal (17), modelo polinomial, etc.. Como se puede observar al igual que en RNA, es necesario definir las dimensiones máximas del modelo.

Este modelo fue propuesto por Leontaritis & Billings y puede tener una gran cantidad de representaciones, cada una con sus ventajas y desventajas dependiendo de la aplicación.

Los coeficientes de la Ec. 17 se obtienen por el método de mínimos cuadrados.

#### RED NEURONAL ARTIFICIAL POLINOMIAL

La historia de las RNA comienza con el trabajo de McCulloch y Pitts (19) planteando algunas ideas para modelar el sistema neuronal. Estos modelos biológicos han cambiado con los nuevos avances reportados en las neurociencias y la tecnología. Actualmente, se sabe que las conexiones sinápticas no solamente pueden

ser modeladas mediante una suma de la ponderación de las entradas (20). Los resultados muestran que algunas neuronas pueden modular, potenciar y ponderar la respuesta de otras neuronas (21). Esto significa que el modelo por neurona pudiera considerarse como una relación de multiplicación o potenciación. En la literatura se pueden encontrar algunos modelos que aprovechan estas ideas como: Las Redes Neuronales Polinomiales (RNP), (Polynomial Neural Networks, PNN) (22), Redes Neuronales de Orden Mayor (Higher Order Neural Networks, HONN) (23) y modelos con interconexiones no lineales. Este tipo de representación no es exclusiva de las RNA y se pueden consultar modelos similares matemáticamente en otras áreas, por ejemplo: el modelo NARMAX (24)(25), el Método de grupo para el manejo de datos (Group Method of Data Handling (GMDH) (26)(27) y las Aproximaciones Polinomiales (28)(29).

El modelo de RNAP propuesto puede ser descrito como:

$$\hat{y}_k = [\phi(x_{1,k}, x_{2,k}, \dots, x_{n_1,k}, x_{1,k-1}, x_{2,k-1}, \dots, x_{n_1,k-n_1}, \dots, y_{k-1}, y_{k-2}, \dots, y_{k-n_2})]_{\phi_{\min}}^{\phi_{\max}} \quad (Ec. 20)$$

donde:  $\hat{y}_k \in \mathfrak{R}$  es el estimado de una función, es decir la salida de la red,  $\phi(x,y) \in \mathfrak{R}$  es una función no lineal,  $x_i \in X$  son las entradas,  $i=1, \dots, n_1$ ;  $n_1$ =número de entradas,  $y_{k-j} \in Y$  son los valores anteriores de la salida,  $j=1, \dots, n_2$ ,  $n_1$  el número de retardos de la entrada,  $n_2$  el número de retardos de la salida,  $X, Y$  son conjuntos compactos de  $\mathfrak{R}$ .

La función no lineal  $\phi(z)$  está dada por:

$$[\phi(z)]_{\phi_{\min}}^{\phi_{\max}} = \begin{cases} \phi_{\max} & \phi(z) \geq \phi_{\max} \\ \phi(z) & \phi_{\min} < \phi(z) < \phi_{\max} \\ \phi_{\min} & \phi(z) \leq \phi_{\min} \end{cases} \quad (Ec. 21)$$

donde  $\phi_{\max}$  and  $\phi_{\min}$  son los límites máximo y mínimo respectivamente.

Para simplificar el manejo de la notación en las ecuaciones se va hacer un cambio de variable de tal forma que:

$$z = \{x_{1,k}, x_{2,k}, \dots, x_{n_1,k}, \dots, y_{k-1}, y_{k-2}, \dots, y_{k-n_2}\} = \{z_1, z_2, z_3, \dots, z_{n_v}\} \quad (Ec. 22)$$

donde:  $n_v$  es el número total de elementos en la descripción  $z$ , es decir, el número total de entradas, valores anteriores y valores anteriores de la salida:

$$n_v = n_1 + n_1 n_1 + n_2 \quad (Ec. 23)$$

Entonces la función  $\phi(z) \in \Phi_p$  pertenece a una familia de polinomios  $\Phi_p$  que pueden ser representados:

$$\Phi_p(z_1, z_2, \dots, z_{n_v}) = \left\{ \begin{aligned} &\phi(z) : \phi(z) = a_0(z_1, z_2, \dots, z_{n_v}) + a_1(z_1, z_2, \dots, z_{n_v}) \\ &+ a_2(z_1, z_2, \dots, z_{n_v}) + \dots + a_p(z_1, z_2, \dots, z_{n_v}) \end{aligned} \right\} \quad (Ec. 24)$$

El subíndice  $p$  es la potencia máxima de la expresión polinomial, en tanto que los términos  $a_i(z_1, z_2, \dots, z_{n_v})$  son polinomios homogéneos de grado total  $i$ , para  $i=0, \dots, p$ . Cada polinomio homogéneo puede ser definido como:

$$\begin{aligned} a_0(z_1, z_2, \dots, z_{n_v}) &= w_0 \\ a_1(z_1, z_2, \dots, z_{n_v}) &= w_{1,1} z_1 + w_{1,2} z_2 + \dots + w_{1,n_v} z_{n_v} \\ a_2(z_1, z_2, \dots, z_{n_v}) &= w_{2,1} z_1^2 + w_{2,2} z_1 z_2 + w_{2,3} z_1 z_3 + \dots \\ &\dots + z_1 z_{n_1} + \dots z_2^2 + \dots z_2 z_3 \dots + w_{2,n_2} z_{n_2}^2 \\ a_3(z_1, z_2, \dots, z_{n_v}) &= w_{3,1} z_1^3 + w_{3,2} z_1^2 z_2 + w_{3,3} z_1^2 z_3 + \\ &w_{3,4} z_1 z_2^2 + w_{3,5} z_1 z_2 z_3 + w_{3,6} z_1 z_3^2 + \dots \\ &+ \dots z_2^3 + \dots z_2^2 z_3 + \dots z_3 z_3^2 + \dots + w_{3,n_3} z_{n_3}^3 \\ a_p(z_1, z_2, \dots, z_{n_v}) &= w_{p,1} z_1^p + w_{p,2} z_1^{p-1} z_2 + \dots \\ &\dots + w_{p,n_p} z_{n_p}^p \end{aligned} \quad (Ec. 25)$$



donde  $w_{ij}$  corresponde al peso asociado a cada neurona. El término  $w_0$  corresponde al *input bias* de la red. Este término tiene el mismo significado que el coeficiente cero de la transformada de Fourier, es decir, obtiene el promedio de la señal que se quiere estimar. El polinomio  $a_1(z)$  corresponde únicamente a la ponderación lineal de las entradas. De  $a_2(z)$  a  $a_p(z)$  se representan los términos de modulación entre las entradas correspondientes y las relaciones de potenciación.

Como se puede observar, los términos utilizados a partir de  $z_i^2$  permiten al algoritmo resolver el problema de paridad bidimensional que tenía el perceptrón. Esto es análogo al efecto de tener varias capas en una red neuronal tradicional.

El valor  $N_i$  corresponde al número de términos de cada polinomio homogéneo:

$$N_0 = 1, N_1 = n_p, N_2 = \sum_{i=1}^n i, N_3 = \sum_{s_1=0}^{n_1-i n_1-s_1} \sum_{i=1} i, \tag{Ec. 26}$$

$$N_p = \underbrace{\sum_{s_{p-2}=0}^{n_p-1} \dots \sum_{s_2=0}^{n_p-s_2} \sum_{s_1=0}^{n_p-s_1} \sum_{i=1} i}_{p-1}$$

La dimensión  $N_\Phi$  de cada familia  $\Phi_p$  puede ser obtenida de la siguiente forma:

$$N_\Phi = \sum_{i=0}^p N_i \tag{Ec. 27}$$

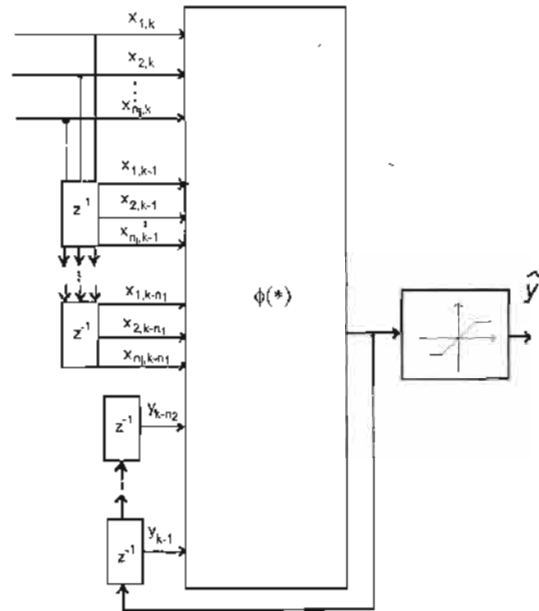


Figura 8 Esquema de RNAP

El modelo ARMAX es uno de los algoritmos más comunes para identificar sistemas dinámicos lineales. La RNAP con el parámetro  $p=1$  puede ser considerada como un modelo ARMAX y con  $p \geq 2$  como un modelo NARMAX. Una de las ventajas de RNAP, es que obtiene la representación óptima y es posible identificar la parte lineal y no lineal de un sistema dinámico. En las siguientes secciones se describe esto en detalle.

#### Aprendizaje de RNAP

Para introducir el aprendizaje en RNAP es necesario, primero, introducir algunos conceptos que serán utilizados en este artículo.

#### Definición 2

El error de aproximación de RNAP se define como:

$$\begin{aligned} err_n(y^n, \phi(z)) &= \frac{1}{n} \sum_{k=1}^n (y_k - \phi(z_k))^2 \\ &= \frac{1}{n} \sum_{k=1}^n (y_k - \hat{y}_k)^2, y^n = (y_1, y_2, \dots, y_n) \end{aligned} \tag{Ec. 28}$$

donde  $y_n$  es la salida deseada,  $\phi(z_k) \in \Phi_p$  y  $n$  es el número de puntos.

Definición 3

El error óptimo está definido como:

$$\begin{aligned} \text{opterr}_n(y^n, \phi(z)) &= \min_{\phi \in \Phi_p} \text{err}_n(y^n, \phi(z)) \\ &= \text{err}_n(y^n, \phi_n^*(z)) \end{aligned} \tag{Ec. 29}$$

donde  $\phi_n^*(z) \in \Phi_p$  es la estimación óptima de  $y^n$ .

Definición 4

La RNAP  $\phi(z) \in \Phi_p$  aprende uniformemente la salida deseada con precisión  $\varepsilon$  si

$$\text{err}_n(y^n, \phi(z)) - \text{err}(y^n, \phi_n^*(z)) > \varepsilon \quad \varepsilon > 0$$

Después de describir estos conceptos ahora el problema del aprendizaje de RNAP consiste en encontrar la estructura de  $\phi \in \Phi_p(z)$  que verifica esta desigualdad.

En la siguiente sección se aplica el uso de algoritmo genético para obtener el valor del arreglo  $W_b^T$ . Se presenta un algoritmo que obtiene la arquitectura óptima de la red mediante el uso de AG (30). Para lograr esto defínase un vector de componentes  $M(z)$ : utilizando la simplificación de (22):

$$\begin{aligned} M(z) &= \{z_1, z_2, z_3, \dots, z_{m_1}, z_1^2, z_1 z_2, \dots \\ &\quad \dots, z_{m_2}^2, z_1^3, z_1^2 z_2, \dots, z_{m_2}^p\} \end{aligned} \tag{Ec. 30}$$

Entonces la función no lineal  $\phi \in \Phi_p(z)$  descrita en la ecuación 24 puede ser representada como:

$$\phi = \langle W, M(z) \rangle \text{ donde } W = W_b \cdot W_b^T, \forall w_i^b \in \{0,1\}$$

(Ec. 31)

donde  $W = \{w_1, w_2, \dots, w_{N_\Phi}\}$  son los pesos de RNAP,  $W_b$  es un vector binario, y  $N_\Phi$  se obtiene utilizando la ecuación 26.

Definición 5

El producto  $\cdot^*$  se define como.

$$W \cdot^* W_b^T = \begin{cases} w_y & \text{if } w_j^b = 1 \\ 0 & \text{if } w_j^b = 0 \end{cases} \tag{Ec. 32}$$

Por ejemplo, para  $W$  y  $W_b$  como:

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \quad W_b = [1 \quad 0 \quad 1]$$

$$W \cdot^* W_b^T = \begin{bmatrix} w_{11} & 0 & w_{13} \\ w_{21} & 0 & w_{23} \\ w_{31} & 0 & w_{33} \end{bmatrix}$$

y para el caso vectorial:

$$[w_1 \quad w_2 \quad w_3 \quad w_4] \cdot^* [1 \quad 0 \quad 1 \quad 0]^T = [w_1 \quad 0 \quad w_3 \quad 0]$$

La ecuación 31 representa que  $\phi$  solamente tiene términos específicos de  $\Phi_p$  que pueden ser seleccionados por  $W_b$  de tal forma que la estructura óptima de RNAP  $\phi^*$  puede ser calculada como:

$$\begin{aligned} \phi^*(z) &= \langle W \cdot^* W_b^T, M(z) \rangle = \langle W \cdot^* (W_b^T)^*, M(z) \rangle \\ &= \langle W, M(z) \cdot^* (W_b^T)^* \rangle \end{aligned}$$

$$\Rightarrow \text{err}_n(y^n, \phi(z)) = \text{err}_n(y^n, \langle (W \cdot^* W_b^T), M(z) \rangle)$$

(Ec. 34)

$$\Rightarrow \underset{W'}{\text{opt err}_n(y^n, \phi(z))} = \underset{W'}{\text{opt err}_n(y^n, \langle\langle W' \rangle\rangle, M(z))}$$

(Ec. 35)

Utilizando las ecuaciones 31 a 35 el problema de aprendizaje para una estructura específica puede ser representado por un problema de optimización con los siguientes dos pasos:

$$\min_{W_b} \min_{W \in \mathbb{R}^N} \text{err}_n(y^n, \phi(z)) \Big|_{W_b} \tag{Ec. 36}$$

donde  $\text{err}_n(y^n, \phi(z)) \Big|_{W_b}$  es el error definido en la ecuación 28 para un valor determinado de  $W_b$ .

Los valores del parámetro  $W$  pueden ser obtenidos utilizando el método de mínimos cuadrados (31):

$$W|_{W_b} = \underset{W \in \mathbb{R}^N}{\text{arg min}} \text{err}_n(y^n, \phi(z)) \Big|_{W_b}$$

$$W|_{W_b} = \Gamma^N \sum_{k=1}^n y_k (M_b(z_k)) \quad M_b(z_k) = M \cdot W_b^T$$

$$\Gamma^N = \left( \sum_{k=1}^n M_b(z_k) M_b(z_k)^T \right)^{-1}$$

(Ec. 37)

o en su forma recurrente:

$$(W|_{W_b})_k = (W|_{W_b})_{k-1} + \Gamma_k M_b(z_k)^T [y_k - (W|_{W_b})_{k-1} M_b(z_k)^T]$$

$$\Gamma_k = \Gamma_{k-1} \cdot \frac{\Gamma_{k-1} M_b(z_k)^T M_b(z_k) \Gamma_{k-1}}{1 + M_b(z_k) \Gamma_{k-1} M_b(z_k)^T}$$

(Ec. 38)

Para este caso  $N=N_\phi$  y el espacio de búsqueda tiene dimensión  $2^N$ . En algunos casos considerando un valor fijo de  $p$ ,  $n_1$ ,  $n_2$  es muy probable que no se requiera de todos los elementos de  $\phi(z)$  y como se describió en la sección anterior es posible seleccionar qué elementos se requieren para obtener una arquitectura o estructura óptima.

De esta forma el problema de aprendizaje puede ser traducido a obtener la óptima estructura de RNAP utilizando AG.

### ALGORITMOS GENÉTICOS

Algoritmo Genético es un modelo de optimización que se encuentra basado en algunos de los mecanismos de evolución que se observan en la naturaleza. Este modelo de computación genética puede ser implementado con el uso de arreglos de bits o caracteres que representan los cromosomas. Aun cuando existen muchos trabajos acerca de cadenas de longitud variable y sobre otras estructuras, la mayoría del trabajo con Algoritmos Genéticos está enfocado hacia cadenas de longitud fija, si no es manejado de esta manera, el Algoritmo de Evolución que se está considerando será Programación Genética, la cual se enfoca hacia cadenas de longitud variable (32,33).

Cuando se implementa el Algoritmo Genético, usualmente sigue el siguiente ciclo (34,35):

- Generación de una población inicial de forma aleatoria.
- Evaluación del más apto o alguna función objetivo de todos los individuos que pertenecen a la población.
- Creación de una nueva población debido a la ejecución de operaciones como recombinación (crossover) y mutación sobre los individuos de donde el más apto o el valor mejorado fue medido.
- Eliminación de la población original e iteración, utilizando la nueva población hasta que el criterio de terminación sea cumplido o se haya llegado a cierto número de generaciones, sin haber completado el objetivo planeado.



Figura 9 Etapas Canónicas del Algoritmo Genético

El distinto tipo de operaciones que se hacen sobre las cadenas o cromosomas que son manipulados dentro del AG son llamadas operadores, en este trabajo se utiliza el operador de recombinación sexual, la mutación (36) y otros procesos especiales que son llamados agregar padres (26). Las siguientes secciones describen estos procesos en detalle.

Recombinación

(Ec. 39)

El operador de recombinación se caracteriza por la combinación de material genético de los individuos que son seleccionados en función del buen desempeño (función objetivo) que tuvieron en comparación con el resto de los individuos o "padres" que conforman la población. Existen algunas variantes de este operador (37), que puede ser determinado por el número de puntos de recombinación que serán considerados para la generación de una población, el número de padres involucrado para la generación del linaje, etc. Es obvio que si el número de puntos de recombinación se incrementa, el número de individuos que conforman el linaje, será también mayor.

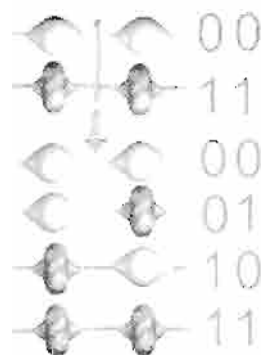


Figura 10 Operador de Recombinación en un Punto.

Para explicar la recombinación multipunto empleado, considérese que  $F_g \subset IB^{n_p \times n_b}$  es el conjunto de padres de una población dada donde  $n_p$  es el número de padres de la población en la generación  $g$  y  $n_b$  es el número de bits del arreglo (cromosoma),  $g=1, \dots, n_g$ ; y  $n_g$  es el número total de generaciones.

$C(F_g, n_r) \in IB^{n_s \times n_b}$  es el operador de recombinación y puede ser definido como la combinación entre el conjunto de padres considerando el número de intervalos  $n_r$  de cada individuo y el número de hijos  $n_s$  por lo que:

$$n_s = n_p^{n_r}$$

Para mostrar cómo es que puede ser aplicado el operador de recombinación, considere que  $F_g$  se forma con  $n_p=2$  y  $n_r=3$ . Esto significa que se divide el arreglo en tres secciones y se denomina a cada sección como  $a_i$  y  $b_i$  respectivamente para  $i=1, \dots, n_r$ . Si:

$$F_g = \begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix}$$

$$\Rightarrow C(F_g, 3) = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_1 & a_2 & b_3 \\ a_1 & b_2 & a_3 \\ a_1 & b_2 & b_3 \\ b_1 & a_2 & a_3 \\ b_1 & a_2 & b_3 \\ b_1 & b_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix}$$

Es importante notar que con este operador los padres  $F_g$  de la población  $g$  se incluyen en el resultado de la recombinación.

Mutación

En el operador de mutación, sólo se niegan algunos bits que son seleccionados en forma aleatoria por medio de un factor de probabilidad  $P_m$ ; en otras palabras, tan sólo se varían los componentes de algunos genes, es decir, se modifican los alelos. Este operador es extremadamente importante, ya que permite asegurar que se mantenga la diversidad dentro de la población, la cual es básica para la evolución (38,39).

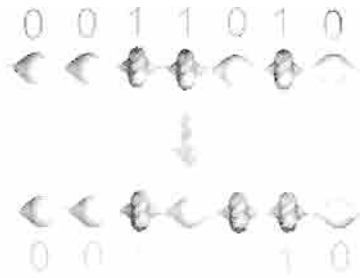


Figura 11 Operador de Mutación

Este operador  $M : IB^{n_r, x n_b} \rightarrow IB^{n_r, x n_b}$  cambia con la probabilidad  $P_m$  la población generada por el operador de recombinación en la siguiente forma :

$$M(C_{ij}) = \begin{cases} C_{ij} \oplus 1 & r \leq P_m \\ C_{ij} & r > P_m \end{cases} \tag{Ec. 40}$$

donde  $r \in U(0,1)$  es una variable aleatoria e,  $i=1, \dots, n_s; j=1, \dots, n_b;$  y  $\oplus$  es el operador *x-or*.

El operador de mutación asegura que la probabilidad de encontrar cualquier punto en el espacio de búsqueda nunca sea cero y evita que se alcance un mínimo local.

**Agregar Padres**

En esta parte sólo se agrega  $F_g$  al resultado del proceso de mutación, entonces la población  $A_g$  en la generación  $g$  se puede obtener como:

$$A_g = \begin{bmatrix} M(C(F_g, n_r)) \\ F_g \end{bmatrix} \tag{Ec. 41}$$

Cabe hacer notar que  $A_g$  tiene los mejores individuos de  $A_{g-1}$  ya que se agrega  $F_g$  en este procedimiento. Este paso y el anterior aseguran la convergencia del algoritmo.

**Proceso de Selección**

El Proceso de Selección  $S_g$  calcula la función objetivo  $O_g$  que representa una función específica que se quiere minimizar y se seleccionan los mejores individuos  $n_p$  de  $A_g$  tal que:

$$S_g(A_g, n_p) = \min^{n_p} O_g(A_g) \tag{Ec. 42}$$

Entonces:

$$F_{g+1} = S_g(A_g, n_p) \tag{Ec. 43}$$

Note que los mejores individuos de la generación  $g$  pueden ser obtenidos por el siguiente operador:

$$S_g(A_g, 1) \tag{Ec. 44}$$

En resumen el Algoritmo Genético puede ser descrito por medio de los siguientes pasos:

1. Para la condición inicial  $g=0$  seleccionar en forma aleatoria  $A_0$ , tal que  $A_0 \in IB^{n_r, x n_b}$
2. Calcular  $F_1 = S_0(A_0)$
3. Obtener  $A_g$
4. Calcular  $S_g$
5. Regresar al paso 3 hasta que el número máximo de generaciones es alcanzado o uno de los individuos de  $S_g$  obtiene el mínimo valor deseado de  $O_g$ .

El empleo de Algoritmo Genético es especialmente útil cuando se tiene gran cantidad de posibilidades en el espacio de búsqueda y no se posee información acerca de cómo obtener la solución óptima para un problema específico. Para este caso, la aplicación de la teoría es automática si se considera a  $W_b^*$  como el arreglo buscado. El problema de aprendizaje puede ser trasladado para obtener la estructura óptima de PANN usando AG. Los pasos de AG modificados pueden observarse en la siguiente tabla 1.

Tabla 1 Pasos de AG y AG en RNAP

ALGORITMO GENÉTICO	ALGORITMO GENÉTICO EN RNAP
1. Para la condición inicial $g=0$ se calcula $A_0$ de forma aleatoria con las siguientes dimensiones $A_0: n_s \times n_p$	1. Para la condición inicial $g=0$ se calcula $A_0$ de forma aleatoria con las siguientes dimensiones $A_0: n_s \times n_p$ donde cada renglón de la matriz corresponde a una propuesta para $W_b$
2. Se obtiene la función objetivo y se selecciona el mejor individuo para la población inicial $F_1 = S_0(A_0, n_p)$	2.1. Para calcular $F_1$ primero se debe calcular $S_0(A_0, n_p)$ donde la función objetivo $O_g$ puede ser calculada utilizando el error definido por (28) de la siguiente forma: $O_0^i = err_n^i(y^n, \langle (W^i)_{W_b}^i, M(z) \rangle), i = 1, \dots, n_s$ 2.2. Se calcula $S_g(A_g, n_p) = \min^{n_p} O_g(A_g)$
3. Se obtiene la nueva población $A_g$ en la generación $g$ con el operador recombinación y mutación.	3. Se obtiene la nueva población $A_g$ en la generación $g$ con el operador recombinación y mutación.
4. Calcular la función objetivo y seleccionar a los mejores individuos de la generación con $S_g$	4. Calcular $S_g$ con la siguiente función objetivo $O_g^i = err_n^i(y^n, \langle (W^i)_{W_b}^i, M(z) \rangle), i = 1, \dots, n_s + n_p$
5. Regresar al paso 3 hasta que se alcance el máximo número de generaciones o uno de los individuos de $S_g$ obtengan el mínimo valor deseado de $O_g$	5. Regresar al paso 3 hasta que se alcance el máximo número de generaciones o uno de los individuos de $S_g$ obtenga el mínimo valor deseado de $O_g$

El error óptimo en la generación  $g$  puede ser obtenido por:

$$\left( \underset{W_b \in \mathbb{R}^{n_p}}{\text{opt } err_n} \right)_g = \min_{W_b \in A_g} err_n \left( y^n, \langle (W^i)_{W_b}^i, M(z) \rangle \right) \quad (\text{Ec. 45})$$

Teorema 1  
Definiendo

$$W_b^g \stackrel{\Delta}{=} S_g(A_g, 1) \quad (\text{Ec. 46})$$

donde  $W_b^g$  es el mejor individuo en la generación  $g$ , es decir, la estructura óptima de la RNAP en ese momento, se puede reescribir el error de aproximación como:

$$err(y^n, \phi_g(z)) = \frac{1}{n} \sum_{k=1}^n \left( y_k - \langle W, M(z) \cdot (W_b^g)^T \rangle \right)^2 \quad (\text{Ec. 47})$$

donde  $y^n$  es la salida deseada,  $\phi_g(z) \in \Phi_p$  es la representación óptima en la generación  $g$ , y

$$W_{W_s}^* = \operatorname{argmin}_{W \in \mathbb{R}^n} \operatorname{err}_n(y^n, \phi(z)) \Big|_{W_s = W_s^*}$$

Si  $P_m > 0$  entonces RNAP aprende uniformemente la salida deseada de tal forma que:

$$\lim_{g \rightarrow \infty} P\{\operatorname{err}(y, \phi_g(z)) - \operatorname{opterr}(y, \phi(z)) \geq \varepsilon\} = 0, \quad \varepsilon > 0$$

(Ec. 48)

$\phi_g \in \Phi_\rho$  puede ser descrita como:

$$\phi_g = \left\langle W, M(z) \cdot * S_g(A_g, 1)^T \right\rangle$$

(Ec. 49)

Debido al progreso de agregar padres y el de mutación

$$\operatorname{err}(y, \phi_{g+1}(z)) \leq \operatorname{err}(y, \phi_g(z))$$

(Ec.50)

$$\Rightarrow \lim_{g \rightarrow \infty} \operatorname{err}(y, \phi_g(z)) = \operatorname{opterr}(y, \phi(z))$$

(Ec. 51)

Comentario 1

Para casos prácticos, el Teorema 1 puede escribirse utilizando la Definición 1 como:

Si  $P_m > 0$  entonces la RNAP aprende uniformemente la salida deseada con precisión  $\varepsilon$  en un número finito de generaciones  $n_g$  si:

$$\lim_{g \rightarrow n_g} P\{\operatorname{err}(y, \phi_g(z)) - \operatorname{opterr}(y, \phi(z)) > \varepsilon\} = 0 \quad \varepsilon > 0$$

(Ec. 52)

Demostración del Teorema 1

Defínase:

$$\Delta_g := \operatorname{err}(y, \phi_{g+1}(z)) - \operatorname{opterr}(y, \phi_g(z))$$

(T1)

Para simplificar la notación se utilizará:

$$\Delta_g := \operatorname{err}^g - \operatorname{opterr}^g$$

(T2)

$$\begin{aligned} \Delta_{g+1} &:= \operatorname{err}^{g+1} - \operatorname{err}^g + \operatorname{err}^g - \operatorname{opterr}^g \\ &= \operatorname{err}^{g+1} - \operatorname{err}^g + \Delta_g \end{aligned}$$



$$\Delta_{g+1} = \Delta_g + \operatorname{err}^{g+1} - \operatorname{err}^g$$

$$\Delta_g = \Delta_{g-1} + \operatorname{err}^g - \operatorname{err}^{g-1}$$

$$\Delta_{g-1} = \Delta_{g-2} + \operatorname{err}^{g-1} - \operatorname{err}^{g-2}$$

$$\Delta_{g-2} = \Delta_{g-3} + \operatorname{err}^{g-2} - \operatorname{err}^{g-3}$$

$$\Delta_1 = \Delta_0 + \operatorname{err}^1 - \operatorname{err}^0$$

$$\begin{aligned} \Delta_{g+1} &= \Delta_0 + \sum_{k=0}^g (\operatorname{err}^{k+1} - \operatorname{err}^k) \\ &= \Delta_0 - \sum_{k=0}^g (\operatorname{err}^k - \operatorname{err}^{k+1}) \end{aligned}$$

(T3)

Defínase:

$$\partial_k := \operatorname{err}^k - \operatorname{err}^{k+1} \geq 0$$

(T4)

Observe que para la prueba del teorema es suficiente probar que:

$$\sum_{k=0}^{\infty} \partial_k \stackrel{a.s.}{=} \infty$$

(T5)

$$S_n(\omega) = \sum_{k=0}^n \partial_k(\omega) = S_{n-1}(\omega) + \partial_n(\omega)$$

(T6)

Para probar (5) es necesario probar que:

$$\sum_{k=0}^{\infty} E\{\partial_k | \mathfrak{S}_{n-1}\}^{a.s.} = \infty \quad (T7)$$

Entonces

$$E\left\{S_n(\omega) | \mathfrak{S}_{n-1}\right\}^{a.s.} = S_{n-1}(\omega) + E\left\{\partial_n(\omega) | \mathfrak{S}_{n-1}\right\} \quad (T8)$$

donde  $\mathfrak{S}_{n-1} = \sigma(\delta_0, \delta_1, \delta_2, \dots, \delta_{n-1})$  es la sigma-álgebra generada en el proceso correspondiente y  $\omega$  es un evento aleatorio. Se tiene :

$$\int \partial_n(\omega) dP(\omega | \mathfrak{S}_{n-1}) = \int_{\omega: \partial_n \geq \varepsilon} \partial_n(\omega) dP(\omega | \mathfrak{S}_{n-1}) + \int_{\omega: \partial_n < \varepsilon} \partial_n(\omega) dP(\omega | \mathfrak{S}_{n-1}) \quad (T9)$$

$$\int \partial_n(\omega) dP(\omega | \mathfrak{S}_{n-1}) \geq \int_{\omega: \partial_n \geq \varepsilon} \partial_n(\omega) dP(\omega | \mathfrak{S}_{n-1}) \geq \varepsilon P\{\partial_n \geq \varepsilon | \mathfrak{S}_{n-1}\} = \varepsilon P_n$$

(T10)

donde  $P_n = P\{\partial_n \geq \varepsilon | \mathfrak{S}_{n-1}\}$

Es importante recordar que para una variable aleatoria  $r(\omega)$  se cumple que el operador mutación es igual a:

$$M(C_{ij}) = \begin{cases} \bar{C}_{ij} & r(\omega) \leq P_m \\ C_{ij} & r(\omega) > P_m \end{cases} \quad y$$

utilizando:

$$P\{A_{s+1} = A_s\} = 0 \quad \forall P_m > 0$$

$$\Rightarrow P\left\{\left( err^s - err^{s+1} \right) \geq \varepsilon | \mathfrak{S}_{n-1}\right\} > 0$$

y considerando

$$err(y, \phi_{g+1}(z)) + \varepsilon \leq err(y, \phi_g(z))$$

entonces:

$$P\{\partial_n \geq \varepsilon | \mathfrak{S}_{n-1}\} \geq p > 0 \quad \forall P_m > 0$$

$$P_n \geq p > 0$$

$$\therefore \sum_{n=1}^{\infty} P_n^{a.s.} = \infty$$

$$\sum_{k=0}^{\infty} \partial_k^{a.s.} = \infty$$

$$\exists g = n_0(\omega) < \infty : \Delta_{n_0(\omega)+1} = 0 \quad \forall \Delta_0$$

$$\Rightarrow \lim_{g \rightarrow n_0(\omega)} err(y, \phi_g(z)) = opterr(y, \phi(z))$$

Q. E. D. . .

Como se puede observar la representación de RNAP puede ser considerada como un caso particular del modelo NARMAX. Por lo tanto, también es posible aplicar la metodología descrita en la sección anterior utilizando AG para este tipo de modelos.

### CONTROL ADAPTABLE CON RNAP

En esta sección se describirá la forma en que una RNAP puede ser implementada para su aplicación en control adaptable de sistemas no lineales utilizando un esquema de control adaptable indirecto.

El modelo de RNAP descrito en el capítulo anterior puede reescribirse como:

$$y_k = \phi_k(x) + \phi_k(y) + \phi_k(x, y)$$

(Ec. 53)

donde  $\phi(x)$  y  $\phi(y)$  representan los términos



lineales de  $\phi(z)$ . Estos términos son equivalentes a tener únicamente el caso de  $\rho=1$ , es decir, el caso para los polinomios

$a_0(z_1, z_2, \dots, z_{n_1})$  y  $a_1(z_1, z_2, \dots, z_{n_1})$   $\phi_k(x, y)$  las relaciones no lineales. Recuerde que:

$$\phi_k(x, y) = \sum a_i(z_1, z_2, \dots, z_{n_1}) \quad (Ec. 54)$$

Si se sustituye en (Ec. 53)

$$y_k - \phi_k(y) = A(q^{-1})y_k \quad (Ec. 55)$$

$$\phi(x_k) = q^{-d}B(q^{-1})x_k \quad (Ec. 56)$$

entonces la ecuación quedaría como:

$$A(q^{-1})y_k = q^{-d}B(q^{-1})x_k + \phi_k(x, y) \quad (Ec. 57)$$

con

$$\begin{aligned} A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{n_1}q^{-n_1} \\ B(q^{-1}) &= b_0 + b_1q^{-1} + \dots + b_{n_1}q^{-n_1} \end{aligned} \quad (Ec. 58)$$

donde los parámetros  $a_i$  y  $b_i$  son desconocidos y serán obtenidos utilizando la teoría descrita. Observe que estos parámetros son los pesos de la RNAP.

Considérese que se cumplen las siguientes suposiciones.

- A.1)  $n_1$  y  $n_2$  son fijos y el retardo  $d$  es conocido
- A.2)  $q^{-d}B(q^{-1})$  tiene todos sus ceros dentro del círculo unitario y  $b_0 \neq 0$

Considérese también que se cumple la siguiente relación:

$$C(q^{-1}) = A(q^{-1})S(q^{-1}) + q^{-d}R(q^{-1}) \quad (Ec. 59)$$

donde:

$$\begin{aligned} C(q^{-1}) &= 1 + c_1q^{-1} + \dots + c_{n_2+d-1}q^{-(n_2+d-1)} \\ S(q^{-1}) &= s_0 + s_1q^{-1} + \dots + s_{d-1}q^{-(d-1)} \\ R(q^{-1}) &= r_0 + r_1q^{-1} + \dots + r_{n_2-1}q^{-(n_2-1)} \end{aligned} \quad (Ec. 60)$$

Con esta identidad polinomial se pueden calcular los valores de  $S(q^{-1})$  y  $R(q^{-1})$  para un valor determinado de  $C(q^{-1})$ .

Si se calcula

$$C(q^{-1})y_i$$

$$\begin{aligned} C(q^{-1})y &= A(q^{-1})S(q^{-1})y + q^{-d}R(q^{-1})y \\ C(q^{-1})y &= S(q^{-1})A(q^{-1})y + q^{-d}R(q^{-1})y \\ C(q^{-1})y &= S(q^{-1})[q^{-d}\{B\lambda + \phi(x, y)\}] + q^{-d}R(q^{-1})y \\ C(q^{-1})y &= q^{-d}[S(q^{-1})\{B(q^{-1})x + \phi(x, y)\} + R(q^{-1})]y = q^{-d}y \end{aligned} \quad (Ec. 61)$$

$$y^* = S(q^{-1})B(q^{-1})x + S(q^{-1})\phi(x, y) + R(q^{-1})y \quad (Ec. 62)$$

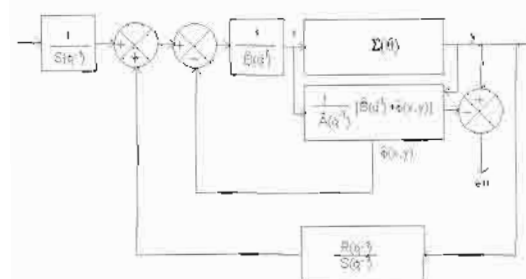


Figura 12 Esquema de Control Adaptable con RNAP

se define la acción de control como:

$$x = X_1 + X_2$$

(Ec. 63)

$$\Rightarrow B(q^{-1})X_1 = -\phi(x,y) \Rightarrow$$

$$S(q^{-1})B(q^{-1})X_2 + R(q^{-1}) = y^*$$

Si el sistema es estable (fase mínima) entonces  $|y_k| < \infty \Rightarrow |x_k| < \infty$

El esquema de control se muestra en la Figura 12. Los parámetros  $A(q^{-1})$ ,  $B(q^{-1})$ , y  $\phi_k(x,y)$  pueden ser estimados por RNAP utilizando AG y posteriormente es posible ajustar la dinámica colocando los polos con  $C(q^{-1})$ . Los siguientes ejemplos muestran la aplicación del algoritmo para el caso lineal y no lineal. Todos los programas utilizados en la simulación fueron desarrollados en Matlab

#### Ejemplo 1 Caso Lineal

Considérese que  $G(z) = \frac{Y(z)}{U(z)}$  es la función de transferencia de un sistema lineal donde:

$$\frac{Y(z)}{U(z)} = \frac{4.9834e^{-3} + 3.3002e^{-5}z^{-1} - 4.9174e^{-3}z^{-2}}{1 - 1.9801z^{-1} + 9.802e^{-1}z^{-2}}$$

(Ec. 64)

La respuesta al escalón se muestra en la Figura 13. En este caso se puede observar que el tiempo de establecimiento del sistema  $t_s=5$  seg. El periodo de muestreo es  $T=0.01$  seg.

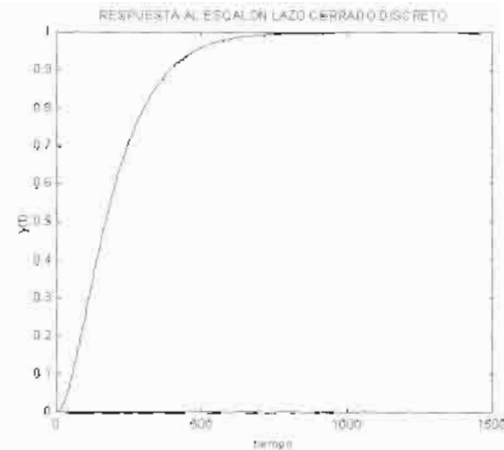


Figura 13 Respuesta al Escalón para Sistema Lineal

El procedimiento para aplicar un control adaptable con RNAP es el siguiente:

1. Identificación de la planta utilizando una señal aleatoria  $u(k)$  con distribución uniforme  $U(-1, 1)$ . Los parámetros utilizados para la red son  $p=2$ ,  $n_1=n_2=2$ ,  $P_m=0.2$ ,  $n_p=3$ ,  $n_r=3$ . Para este ejemplo este paso es equivalente a obtener el modelo simplificado utilizando herramientas tradicionales del área de identificación y control.
2. Probar los resultados utilizando otra entrada como  $r=\text{sen}(t)$ .
3. Separar las partes lineal y no lineal del modelo para obtener  $A(q^{-1})$ ,  $B(q^{-1})$  y  $\phi_k(x,y)$  respectivamente para cancelar la dinámica no lineal y adaptar la dinámica lineal del sistema. Como se puede observar en este caso  $\phi(x,y)=0$ , esto significa que no existe parte no lineal y los resultados se obtienen aplicando únicamente un controlador adaptable típico. Escogiendo los siguientes valores para C:

$$C(q^{-1}) = 1 - 1.9025z^{-1} + 0.9048z^{-2}$$

(Ec. 65)

se escogió una dinámica que tenga un tiempo de establecimiento igual a  $t_s=1$  seg.

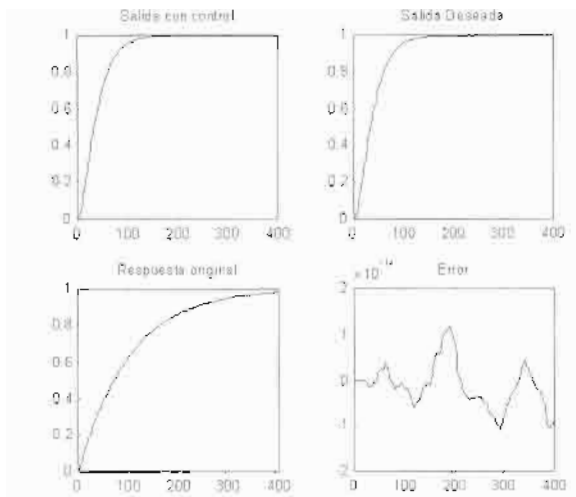


Figura 14 Respuesta con Controlador

La Figura 14 muestra la respuesta original, la salida deseada y la salida con el controlador. Como se puede observar la respuesta al controlador tiene un tiempo de establecimiento de 1 seg., tal y como se definió con el valor de C. De igual forma el error entre la salida deseada y la salida del sistema con el control es muy pequeño.

Ejemplo 2 Caso No lineal

En este caso se va a utilizar un modelo no lineal matemáticamente muy simple propuesto por May (40). Este modelo tiene la siguiente representación:

$$y_k = r(1 - y_{k-1}^2) \tag{Ec. 66}$$

Con  $r=1.2$  y  $y_0=0.5$  se obtiene una dinámica caótica sin el uso de algún tipo de entrada. La serie de tiempo generada por esta ecuación puede observarse en la Figura 15.

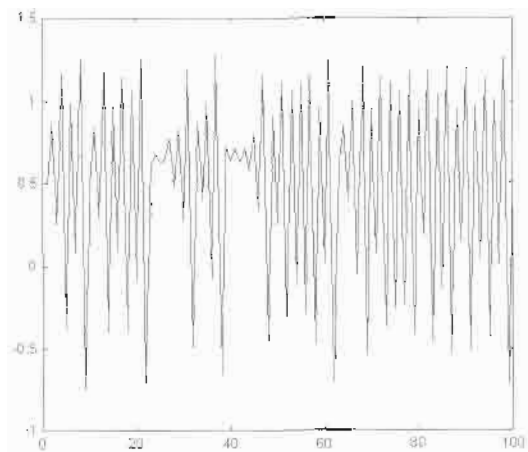


Figura 15 Serie de May

El modelo que se propone como entrada para este sistema es el siguiente:

$$y_k = r(1 - y_{k-1}^2) + u_k \tag{Ec. 67}$$

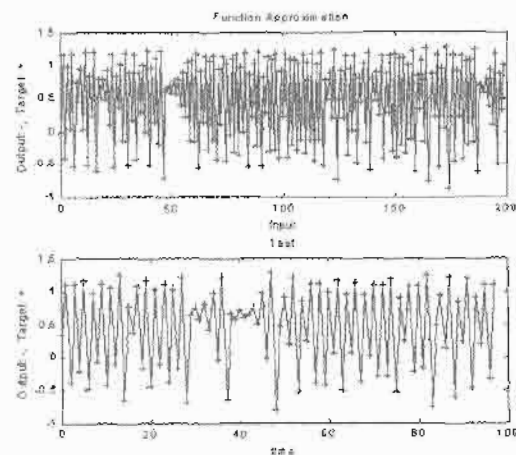


Figura 16 Serie Identificada por RNAP

El proceso de identificación se muestra en la Figura 16 utilizando los mismos parámetros para la RNAP que en el ejemplo anterior. La única diferencia es el rango de valores de la señal de entrada  $u \in [-0.1, 0.1]$ . Separando la parte lineal de la no lineal en la misma forma que en el ejemplo previo, con el mismo valor para C, se puede observar en la Figura 17 el resultado de la metodología propuesta.

La gráfica de respuesta al escalón (salida con control) es la misma que el sistema lineal obtenido en el ejemplo anterior con un tiempo de establecimiento de 1 segundo.

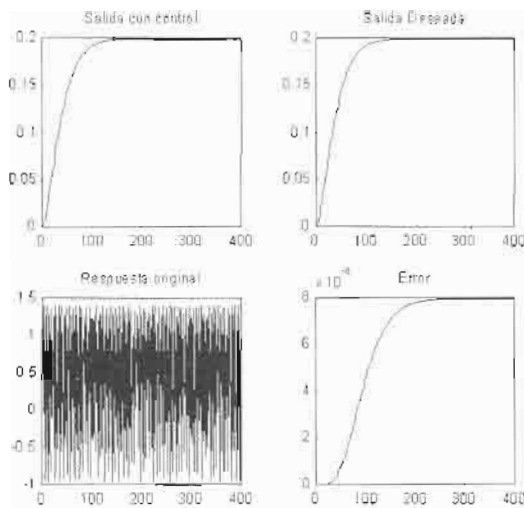


Figura 17 Respuesta con Controlador

### Discusión de Resultados

Como se puede observar en los ejemplos anteriores la RNAP es capaz de generar los parámetros para identificar un sistema no lineal. La combinación de la red neuronal con algoritmo genético permite extraer la información y las relaciones que existen entre los datos de los patrones de entrenamiento, optimizando el error y obtenido la estructura óptima de la red. El modelo utilizado de algoritmo genético permite asegurar la convergencia del proceso utilizando la mutación y el procedimiento de agregar padres.

Si se utiliza la RNAP para sistemas lineales el resultado es un modelo *simplificado* utilizando únicamente los elementos del modelo. Esto es equivalente a obtener la versión óptima del modelo ARMAX.

La naturaleza de la RNAP permite que hacer una separación de la parte lineal y no lineal del sistema o planta. La parte no lineal identificada se utiliza para cancelarla del sistema dinámico original y se aplica un controlador adaptable li-

neal para ajustar la dinámica de la planta "linealizada". El éxito del control adaptable radica principalmente en que el proceso de identificación aproxime lo mejor posible la dinámica no lineal de la planta. En este caso se utilizó un esquema de fase mínima pero la aplicación a sistemas de fase no mínima es similar usando la descripción de la Sección de Control de Sistemas Lineales de Fase no mínima en Tiempo Discreto. Por ser un control adaptable indirecto la ley de control utilizada puede ser cualquiera que cumpla con los requerimientos de desempeño deseado.

### REFERENCIAS

- (1) Antsaklis, P. J., Intelligent Learning Control, Control Systems, vol. 15, núm. 3, Junio 1995.
- (2) Cox, E., Fuzzy Fundamentals, *IEEE Spectrum*. USA, octubre, 1992.
- (3) Gupta, M. & Rao D., Neuro-Control Systems. A Tutorial. *Neuro-Control Systems, Theory and Applications*. IEEE Press. 1994
- (4) Landau, I. D., R. Lozano & M. Saad, *Adaptive Control*. Springer Verlag, 1998.
- (5) Levin, A. & S. Narendra, Control of Nonlinear Dynamical Systems Using Neural Networks: Controlability and Stabilization. *IEEE Transactions on Neural Networks* vol. 4, núm. 2, marzo 1991
- (6) Phan, D. & S. Oh, Adaptive Control of Dynamic Systems using Neural Networks. *Proceedings of IEEE International Conference on System Man and Cybernetics*, 1993.
- (7) Funahashi, K., On the Approximate Realization of Continuous Mappings by Neural Networks. *Neural Networks*. vol. 2, pp. 283-192, 1989.
- (8) Hornik, K., M. Stinchcombe & H. White, Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks*. vol. 2, pp. 359-366, 1989.
- (9) Cybenko, G., Approximation by Superpositions of Sigmoidal Function. *Mathematics of Control Signals and System* vol. 2, pp. 303-314, 1989.
- (10) Cotter, N., The Stone-Weierstrass Theorem and Its Application to Neural Networks. *IEEE Transactions on Neural Networks* vol. 1, núm. 4, pp. 290-295, diciembre 1990.
- (11) Girosi, F. & T. Poggio, Networks and the Best Approximation Property. *Biological Cybernetics*, 1990.

- (12) Narendra, K. & K. Parthasarathy, Identification and Control of Dynamical Systems Using Neural Networks *IEEE Transactions on Neural Networks*, vol. 1, núm. 1 pp. 4-7, marzo 1990.
- (13) Chen, S. & S. Billings, Representations of nonlinear systems: the NARMAX model, *Int. J. Control*, vol. 49, núm. 3, 1989.
- (14) Leontaritis, Y. & S. Billings, Input-output parametric models for nonlinear systems, *Int. J. Control*, vol. 41, núm. 2, 1985.
- (15) Narendra, K. & S. Mukhopadhyay, Adaptive Control using Neural Networks and Approximate Models, *IEEE Transactions on Neural Network*, vol. 8, núm. 3, pp. 475-485, mayo 1997.
- (16) Chen, S. & A. Billings, Representations of non-linear systems: the NARMAX model, *Int. J. Control*, vol. 49, núm. 3, pp. 1013-1032, 1989.
- (17) Chen, S. & A. Billings, Neural Networks for nonlinear dynamic system modelling and identification, *Int. J. Control*, vol. 56, núm. 2, pp. 319-346, 1992.
- (18) Leontaritis, I. J. & S.A. Billings, Input-output parametric models for non-linear systems. Part I. Deterministic non-linear systems, *Int. J. Control*, vol. 41, núm. 2, pp. 303-328, 1985.
- (19) MacCulloch, Warren S. & Walter, Pitts, A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5:115-133, 1943
- (20) MacGregor, R. J., *Neural and Brain Modeling*. Academic Press, San Diego, Ca, 1987.
- (21) Hornik, K., M. Stinchcombe & H. White, Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks*, vol. 2, pp. 359-366, 1989.
- (22) Park, D.C. et al., "Electric Load Forecasting Using an Artificial Neural Network", *IEEE Trans. Power Systems*, vol. 6, núm. 2, pp. 442-449, 1991.
- (23) Chang, C., J. Lin & J. Cheung, Polynomial and Standard Higher Order Neural Network. *IEEE International Conference on Neural Networks*, pp. 989-994, vol. 2, San Francisco, CA, USA, 28 marzo-abril, 1993.
- (24) Alippi, C. & V. Piuri, Experimental Neural Networks for Prediction and Identification, *IEEE Transactions on Instrumentation and measurement*, vol. 45, núm. 2, abril 1996
- (25) Chen, S. & A. Billings, Recursive Prediction error parameter estimator for nonlinear models, *Int. J. Control*, vol. 49, núm. 2, 1989.
- (26) Ivakhnenko, A., Polynomial Theory of Complex systems, *IEEE Transactions on systems, Man and Cybernetics*, vol. SMC-1, núm. 4, octubre 1971.
- (27) Duffy, J. & M. Franklin, A Learning Identification Algorithm and Its Application to an Environmental System, *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-5, núm. 2, marzo 1975.
- (28) Shin, Y. & J. Gosh, Approximation of Multivariate Functions Using Ridge Polynomial Networks, *IJCNN'92*, pp. 380-5, Baltimore, MD, USA, 7-11 junio, 1992
- (29) Shin, Y., Modified Bernstein Polynomials and Their Connectionist Interpretation, *ICNN*, pp. 1433-8, vol. 3, Orlando, Florida, USA, 27 junio-julio, 1994.
- (30) Gómez-Ramírez, E., A. Poznyak, A. González Yunes & M. Avila-Alvarez, Adaptive Architecture of Polynomial Artificial Neural Network to Forecast Nonlinear Time Series. *CEC99 Special Session on Time Series Prediction*. Mayflower Hotel, Washington D.C., USA, julio 6-9, 1999.
- (31) Gómez Ramírez E. & A. Poznyak, How to Select a Number of Nodes in Artificial Neural Networks. *Neural Networks Applied to Control and Image Processing. NNACIP'94*, CINVESTAV IPN, Noviembre 7-11, Ciudad de México, 1994.
- (32) Altenberg, L., *The Evolution of Evolvability in Genetic Programming*, MIT Press, 1994.
- (33) Andre, D., *Automatically Defined Features: The Simultaneous Evolution of 2-Dimensional Feature Detectors and an Algorithm for Using Them*, MIT Press, Kenneth E., 1994
- (34) Alander, J. T., *An Indexed Bibliography of Genetic Algorithms: Years 1957-1993*, 1994, Art of CAD ltd.
- (35) Bedner, I., *Genetic Algorithms and Genetic Programming at Stanford 1997*, Stanford Bookstore
- (36) Andre, D., J. Koza, *Advances in Genetic Programming 2*, MIT Press, 1996.
- (37) Andrews, M. & R. Prager, *Advances in Genetic Programming*, MIT Press, 1994.
- (38) Altenberg, L., *Genome growth and the evolution of the genotype-phenotype map*, Springer-Verlag, Berlin, Alemania, 1995
- (39) Banzhaf, W., Nordin P, Keller R & Francone F., *Genetic Programming -- An Introduction On the Automatic Evolution of Computer Programs and its Applications*, Morgan Kaufmann, dpunkt verlag, 1997.
- (40) May, R.M., *Nature* (London), vol. 261, p. 459, junio 1976.