

Alarma Predictiva Mediante Reconocimiento de Patrones y Optimización

Luis Pastor Sánchez Fernández

Centro de Investigación en Computación del IPN

Av. Juan de Dios Bátiz esquina con M. Othón de Mendizábal

Unidad Profesional Adolfo López Mateos. México, D.F. 07738, México

lsanchez@cic.ipn.mx

RESUMEN

Con el objetivo de supervisar variables de procesos, se utilizan modelos continuos de 1er. orden y 2do. orden críticamente amortiguado, sobreamortiguado y subamortiguado, para el ajuste de curvas. Sus parámetros son identificados mediante un algoritmo de optimización. Antes de realizar la identificación se determina mediante una red neuronal (RN) backpropagation, previamente entrenada, cual será el modelo más apropiado, lo cual disminuye el tiempo total de procesamiento.

La RN de tres capas, tiene 30 neuronas de entrada, 11 en capa oculta y 4 neuronas de salida. Se utiliza un almacenador circular para guardar los n últimos valores adquiridos de cada variable.

Antes de ejecutar la RN, los datos almacenados son acondicionados y filtrados digitalmente. Posteriormente se interpola o se diezma para obtener 30 puntos. La salida de RN indicará cual es el modelo más apropiado. Finalmente, mediante un algoritmo de ajuste de curvas, se estiman los parámetros de dicho modelo.

Palabras clave: alarma, identificación, neuronales, predicción, redes, supervisión.

INTRODUCCIÓN

Como antecedentes existen métodos y algoritmos de alarmas para la supervisión de procesos tecnológicos y dispositivos que permiten detectar condiciones especiales o anormales. Estas condiciones serán determinadas, en general, por los valores que tomen las variables de dicho proceso [1]. Son aplicables algoritmos de alarma por límites y por histéresis, pero están limitados a diagnosticar condicionales que ya existen o que están muy próximas a ocurrir. Es objetivo del presente trabajo desarrollar algoritmos más elaborados usando modelos matemáticos que representen la dinámica de los procesos que serán supervisados [2]. El método que se presenta permite predecir, con suficiente tiempo, posibles condiciones anormales, lo que propiciará realizar un conjunto de acciones preventivas para evitar tales modos de operación del sistema o realizar un conjunto de acciones para el funcionamiento exitoso del proceso cuando se llegue al estado crítico que no necesariamente tiene que ser

anormal, como es el caso de la operación de los canales de riego cuya dinámica presenta un gran retardo de tiempo.

Las RN son herramientas matemáticas muy poderosas en la identificación de sistemas. La red backpropagation es utilizada de forma muy exitosa para reconocer patrones diversos y en el trabajo que se presenta se utiliza para reconocer patrones de señales de sistemas dinámicos de 1er. orden y de 2do. orden [1], con los cuales se puede representar, con buena aproximación, la dinámica de una considerable cantidad de procesos tecnológicos. La metodología utilizada consiste en estimar los parámetros de los modelos mediante un algoritmo de optimización [3][4]. Antes de realizar la estimación de dichos parámetros se determina, mediante una red neuronal backpropagation previamente entrenada, cual será el modelo más apropiado, lo cual disminuye el tiempo total de procesamiento.

II. DIAGRAMA DE BLOQUES DEL MÉTODO

En la fig. 1. se presenta el diagrama de flujo simplificado de un ciclo del algoritmo de alarma predictiva. Se utiliza un almacenador circular de dimensión configurable. Este ciclo inicia con el almacenamiento permanente de los N últimos datos de la variable del proceso tecnológico o dispositivo que es supervisado. Mediante un algoritmo de por tendencia lineal, como se ilustra en la fig. 2, se determina el instante en el cual se debe iniciar el proceso de reconocimiento del patrón de la señal, correspondiente a los puntos guardados en el almacenador circular. En esta predicción se considera que el comportamiento de la variable mantendrá una tendencia lineal y los parámetros de la ecuación son determinados a partir de la información en el instante de muestreo anterior y actual como se presenta en el gráfico de una variable contra el tiempo, donde LSA es el límite superior de alarma, $v(k)$ y $v(k-1)$ los valores de la variable en los instantes de muestreo actual y anterior, respectivamente y T el intervalo de muestreo.

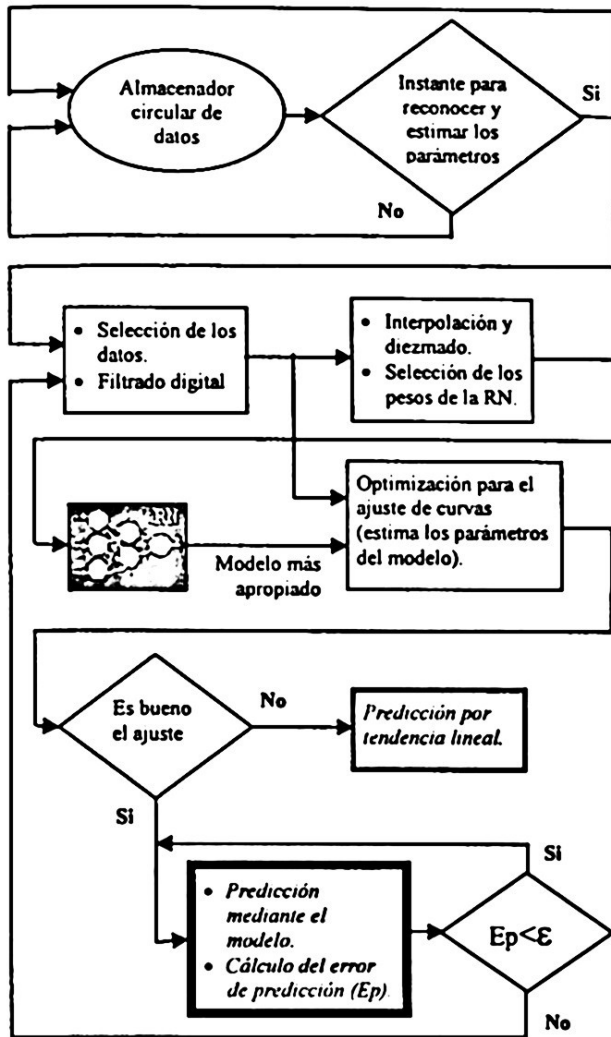


Fig. 1. Diagrama de flujo de un ciclo del algoritmo de alarma predictiva.

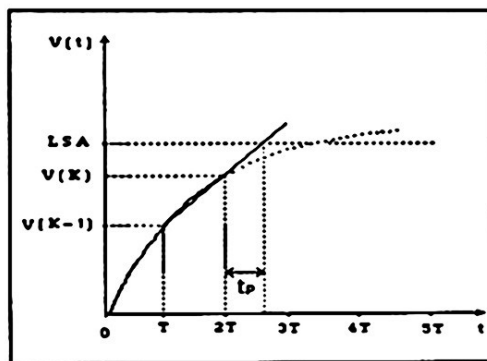


Fig. 2. Predicción basada en la tendencia lineal.

De la fig 2. se puede plantear que:

$$\frac{V(k) - V(k-1)}{2T - T} = \frac{LSA - V(k)}{tp} \quad (1)$$

Obteniendo tp como:

$$tp = \left(\frac{T}{V(k) - V(k-1)} \right) [LSA - V(k)] \quad (2)$$

Se debe establecer el tiempo máximo de pronóstico T_{mp} , de modo que si $tp < T_{mp}$, se inicia el proceso de reconocimiento del patrón de la señal representada por las muestras guardadas en el almacenador circular.

Si es el instante de reconocer y estimar los parámetros del modelo, se seleccionan los datos como se ilustra de forma simplificada en la fig. 3 desechándose los datos más viejos hasta el punto donde se produce un cambio en la tendencia de la pendiente, como se señala con los puntos 1, 2, 3 y 4. Posteriormente se realiza un filtrado con un filtro de media móvil [5]. Se realiza una conversión de la frecuencia de muestreo combinando la interpolación y el diezmado lo cual permite obtener 30 puntos. Se selecciona el conjunto de los pesos de la RN, de acuerdo al signo de la pendiente de la curva formada por los 30 puntos pues se entrenó la misma para patrones con pendiente positiva y negativa. La RN dará como salida el modelo más apropiado para el cual se ajustarán el total de los puntos seleccionados. El algoritmo de optimización retorna un índice minimizado, si éste es menor que un valor preestablecido como aceptable, se considera que el ajuste (estimación de los parámetros del modelo) fue bueno y se utilizará el modelo para predecir el tiempo en el cual la variable alcanzará el valor límite. Como índice de optimización se toma el siguiente:

$$I = \sum_{i=1}^N k_i [V(i) - V_m(i)]^2$$

Siendo:

I : Índice que debe ser minimizado.

$V(i)$: valor de la variable en el instante de muestreo i .

$V_m(i)$: valor de la salida del modelo en el instante de muestreo i .

N : número de muestras de la variable utilizados para la identificación o estimación de los parámetros.

k_i : coeficientes de peso (se está usando valor 1).

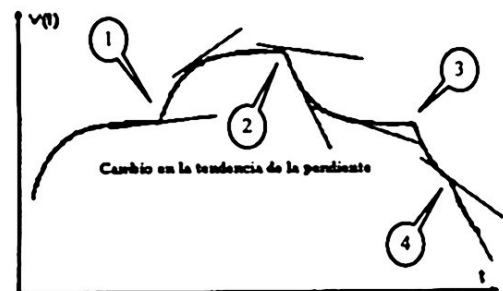


Fig. 3. Ilustración del procedimiento para tomar los puntos que se utilizarán en la identificación.

Para cada pronóstico se calcula un error de predicción aproximado Ep , si es menor que un valor ϵ preestablecido ($Ep < \epsilon$), se continua la predicción mediante el modelo, en caso contrario, como se ha continuado guardando puntos en el almacenador circular, se realiza otro proceso de estimación de

parámetros.

Si la estimación o ajuste no fue bueno se realiza la predicción mediante un algoritmo de tendencia lineal. El error de predicción para un sistema de primer orden [1] se calcula considerando lo siguiente:

$$V(t) = A(1 - e^{-\frac{(t-\theta)}{T_1}})$$

Se obtiene la derivada respecto al tiempo

$$\frac{dV(t)}{dt} = \frac{A}{T_1} e^{-\frac{(t-\theta)}{T_1}}$$

y se evalúa la expresión para $t = kT - 0.5T$.

Adicionalmente, en cada instante de actualización de la alarma, se puede aproximar la derivada de la respuesta de la variable en tiempo real como:

$$\frac{\Delta V(t)}{\Delta t} = \frac{V(k) - V(k-1)}{T}$$

Se compara el resultado de las derivadas obtenidas:

$$\text{Si } \left| \frac{dV(t)}{dt} - \frac{\Delta V(t)}{\Delta t} \right| \leq \varepsilon$$

Se considera que el modelo continúa siendo representativo del proceso, siendo innecesaria una nueva identificación. Para ε el criterio es el siguiente:

$$\varepsilon = ETI$$

Donde:

I: índice que es minimizado durante la identificación de los parámetros del modelo.

T: período de muestreo de la variable.

E: constante que debe ser ajustada por el operador. Un valor pequeño implica una mayor frecuencia en la estimación de los parámetros del modelo obteniéndose mejor exactitud en la predicción a costa de utilizar un mayor tiempo de procesamiento.

Para sistemas de segundo orden, al iniciarse la predicción mediante el modelo, se calcula el tiempo en el cual la variable alcanzará el valor límite, ese tiempo se actualiza en cada período de muestreo. Si la exactitud en la predicción fuera 100%, al tiempo inicial calculado sólo se le restaría el período de muestreo. Para obtener el error de predicción, se presenta a continuación, de forma simplificada, el procedimiento:

```

Deviation:=Abs(RealValue-ModelValue);
DeltaModelValue:=Abs(PreviousModelValue-ModelValue);
IF (Deviation>0.5*DeltaModelValue) THEN BEGIN
  IF PreviousError THEN Dec(CounterRepError)
  ELSE CounterRepError:=1;
  PreviousError:=true;
  IF CounterRepError=0 THEN BEGIN
    PreviousError:=False;
    CounterRepError:=NumRepError;
    ActivateAlarmPred:=False;
    CounterTendencySuc:=1;
    IdentRealized:=False;
    Exit;
  END;
END;
DeltaTime:=( Deviation *SamplingTime)/

```

DeltaModelValue;

DeltaTime:= DeltaTime * Slope;

Time:= DeltaTime + SamplingTime;

PredictionTime:= InitPredictionTime-Time;

En otras palabras, en cada instante de muestreo, se calcula la desviación entre el valor real medido y el calculado mediante el modelo. Esta desviación se utiliza para realizar la corrección. Si la desviación (DeltaTime) es cero, entonces al tiempo de predicción calculado inicialmente mediante el modelo (InitPredictionTime), se le restaría solo el período de muestreo (SamplingTime). Si la desviación (Deviation) es mayor que el 50% de la diferencia entre los dos valores consecutivos calculados mediante el modelo (DeltaModelValue) y ese comportamiento se repite un número de veces consecutivas (NumRepError), entonces se considera que la exactitud en la predicción es mala ($Ep < \varepsilon$ en la fig. 1) y se reinicia el proceso de estimación de los parámetros de un modelo. El procedimiento explicado anteriormente también se puede utilizar para sistema de primer orden.

2.1 Algoritmo de alarma predictiva por tendencia lineal

1. Hallar el valor V_x que se alcanzaría en un tiempo igual a T_{mp} con la *tendencia lineal presente en el instante de muestreo actual*. Se realiza sustituyendo en la fórmula (2) el LSA por V_x y t_p por T_{mp} , obteniéndose:

$$V_x = \left[\frac{V(k) - V(k-1)}{T} \right] T_{mp} + V(k) \quad (3)$$

2. Determinar la gama a la que pertenece el valor V_x , llamándole *GamaTendencia*:

IF $V_x > LSA$ THEN

GamaTendencia = Alta

ELSE IF $V_x < LIA$ THEN *GamaTendencia* = Baja

ELSE *GamaTendencia* = Normal

3. Determinar la gama del valor de la variable en el instante de muestreo actual $V(k)$:

IF $V(k) > LSA$ THEN *GamaActual* = Alta

ELSE IF $V(k) < LIA$ THEN *GamaActual* = Baja

ELSE *GamaActual* = Normal

4. IF *GamaTendencia* $\neq$ *GamaActual* THEN

{Hay que señalizar y se debe determinar cual es el límite de alarma (LA) que será alcanzado}

CASE *GamaTendencia* OF

Alta : LA = LSA

Baja : LA = LIA

ELSE IF $V(k-1) > LSA$ THEN LA = LSA

ELSE LA = LIA.

Pasar al punto 6.

5. Si *GamaTendencia* = *GamaActual* (no ocurrió la condición del punto 4) entonces *no es necesario estimar los parámetros del modelo pues la tendencia no lleva a la variable, en un tiempo menor o igual que T_{mp} , a un cambio de gama y se pasa al punto 7*.

6. Se calcula t_p utilizando la ecuación (2) sustituyendo LSA por LA:

$$tp = \left(\frac{T}{V(k) - V(k-1)} \right) [LA - V(k)] \quad (4)$$

7. Se hace $V(k-1) = V(k)$ para el próximo instante de muestreo.

III. RECONOCIMIENTO DE PATRONES

Se programó una RN backpropagation en una DLL que es utilizada desde el LabVIEW [6]. La selección de los patrones de entrenamiento, se basó en el comportamiento de las respuestas dinámicas de los sistemas de primero y segundo orden a estímulos paso escalón por ser los más frecuentes. En la fig. 4 se muestran las respuestas de un sistema de 2do. orden críticamente amortiguado, con frecuencias naturales de oscilación w_n igual a 1, 0.5 y 0.25, respectivamente. Para cada curva se muestran 30 puntos, los cuales han sido tomados a frecuencias de muestreo aproximadas de 4, 2 y 1 muestras por segundo, respectivamente. Es por ello, que cada intervalo de tiempo en el eje X, será el periodo de muestreo de cada curva. Si los puntos de las tres curvas son graficados usando un mismo intervalo de tiempo para el eje X, quedan superpuestas como se ilustra en la fig. 5.

Similar comportamiento se presentará en sistemas de primer orden respecto a la constante de tiempo, y en sistemas de segundo sobreamortiguados y subamortiguados [1], en los cuales sólo habrá diferencia respecto a su coeficiente de amortiguamiento ζ , como se ilustra en la fig. 6.

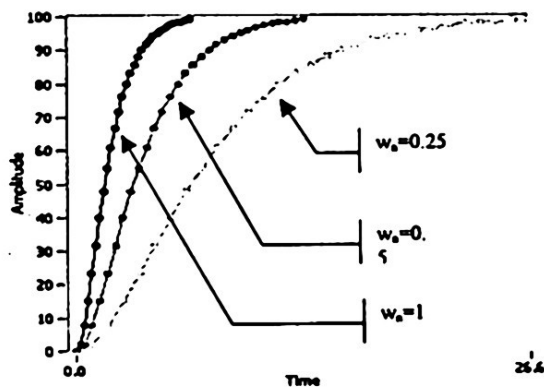


Fig. 4. Sistema críticamente amortiguado.

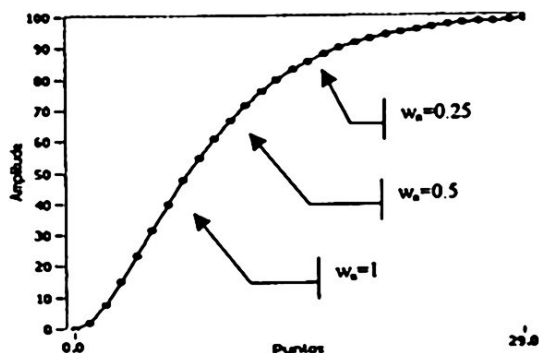


Fig. 5 Las tres curvas de la fig. 4, superpuestas.

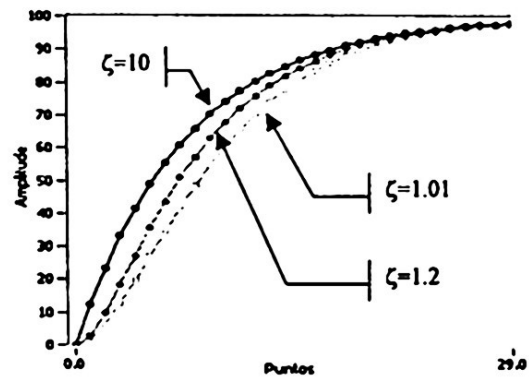


Fig. 6 Respuestas sobreamortiguadas.

Se observa que la w_n no tiene influencia en el patrón de la señal cuando cada marca en el eje X es el orden en que se tomaron los puntos, en lugar del tiempo. No es de interés estimar los parámetros del modelo con la RN (constante de tiempo y ganancia para 1er. orden; ganancia, ζ y w_n para sistemas de 2do. orden) y sí reconocer cual es el tipo de modelo que mejor ajusta o representa al registro de datos almacenados. Por tanto para los patrones de entrada de sistemas de 1er. orden y 2do. críticamente amortiguado sólo se varía la amplitud, para 2do. orden sobreamortiguado y subamortiguado sólo se varían la amplitud y el ζ . Las variaciones en amplitud se toman en %, normalizadas, desde el 40% al 90%. Cuando el sistema está en operación y corresponde procesar la señal, si el valor máximo de la misma está entre 40 y 90 se deja con su valor; si es inferior a 40, se amplifica para llevarla hasta 40; Si es superior a 90, se atenúa para reducirla a 90. Se observaron mejores resultados cuando se amplifica o atenúa, según el caso, hasta llevarla a 90, siempre que la misma esté fuera del intervalo de 40-90. Después de numerosas pruebas, se realizó el entrenamiento con 858 patrones de entrada, distribuidos de la manera siguiente:

- Para sistemas de 2do. orden sobreamortiguado (SSobre):
Por cada valor de ζ se obtienen 11 patrones correspondientes a las variaciones de la amplitud desde 40 hasta 90, con incremento de 5 (40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90). El ζ se varía desde 1.2 hasta 3, con incremento de 0.09, obteniéndose un total de 220 patrones. Para ζ mayores que 3, el sistema se confunde con uno de 1er. orden.
- Para sistemas de 2do. orden subamortiguado (SSub):
De forma similar por cada valor de ζ se obtienen 11 patrones correspondientes a las variaciones de la amplitud. El ζ se varía desde 0.1 hasta 0.7, con incremento de 0.0667, obteniéndose un total de 99 patrones.
- Para sistemas de 1er. orden (P) y 2do. orden críticamente amortiguado (SC) se crean 11

patrones, respectivamente, correspondientes a variaciones de la amplitud desde 40 hasta 90.

Para tener un número similar de patrones para cada modelo y lograr un mejor entrenamiento de la RN, los patrones de (P) y de (SC) se repiten 20 veces, respectivamente, para un total de 440 patrones. Para (SSub.) se repiten dos veces para 198 patrones. EN TOTAL 858 PATRONES. Se obtienen muy buenos resultados en el entrenamiento y en la generalización de la RN.

Una vez seleccionados los patrones, se fueron utilizando diversas topologías hasta obtener la más simple pero que diera una respuesta adecuada. Finalmente se utilizó una red neuronal con 30 entradas, 11 neuronas en la capa oculta y cuatro neuronas de salida.

IV. EVALUACIÓN DE LOS RESULTADOS

En la tabla 1 se presentan resultados con el modelo de primer orden, en un canal Rehbock de investigaciones de obras hidráulicas, de 20 m de longitud, 0.85 m de ancho y 1.10 m de alto, con compuertas al final y en el centro y válvula de entrada de 12 pulgadas de diámetro. En la fig. 7 se muestran las respuestas temporales del nivel antes de la compuerta central. Se ha calculado el error que se comete con la predicción por tendencia lineal (E_{PL}) y exponencial (E_{PE}), utilizándose las expresiones siguientes:

$$EPL = \frac{|T_{real} - T_s|}{T_{real}} \times 100; \quad EPE = \frac{|T_{real} - t_p|}{T_{real}} \times 100$$

En la tabla 1: LSA es el límite superior de alarma; T es el periodo de muestreo; T_s significa el tiempo de señalización (tiempo mínimo de pronóstico que desea el operador, el cual es utilizado por la predicción por tendencia lineal para determinar si es el momento de iniciar la estimación de los parámetros del modelo); T_p es el tiempo de predicción (tiempo pronosticado para alcanzar el valor límite); T_{real} representa el tiempo para el cual se alcanzó el valor límite.

Resultados similares se obtuvieron cuando se utilizan modelos de segundo orden.

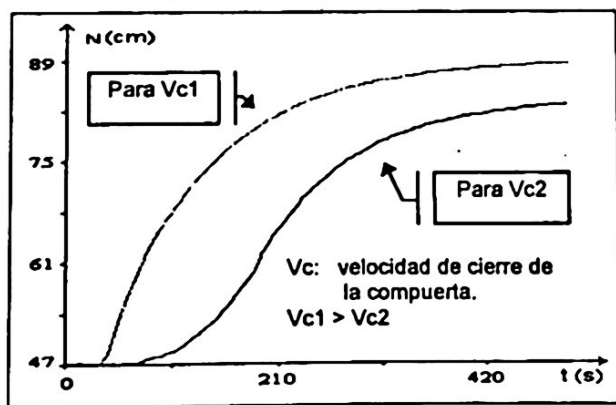


Fig. 7. Nivel antes de la compuerta intermedia.

Tabla 1. Evaluación con modelo de primer orden.

Resp.	LSA cm	T seg	T_s seg	T_p seg	T_{real} seg	EPL %	EPE %
Vc1	80	4	80	140.8	140	43	0.6
Vc1	81	4	80	160	156	49	2.5
Vc2	80	4	50	44.19	44	14	0.4
Vc2	80	4	60	76	80	25	5
Vc2	81	4	60	71	68	13	4
Vc2	81	4	65	89	84	23	5

V. CONCLUSIONES

Se obtuvieron resultados satisfactorios en el entrenamiento de la RN, con alto nivel de generalización. Durante la operación reconoció todas las señales utilizadas, incluso las afectadas por ruidos. Ha significado un paso de avance satisfactorio para el desarrollo de algoritmos eficientes de alarma predictiva por tendencia con un consumo mínimo de tiempo de procesamiento. Asegura la actualización del mensaje de predicción, en cada instante de alarma, lográndose que el error de predicción aparezca distribuido en el eje del tiempo. Actualmente el método ha sido aplicado utilizando el LabVIEW de National Instruments y DLL escritas en otros lenguajes de programación como C y DELPHI.

Como trabajos inmediatos y futuros se están utilizando señales afectadas por niveles de ruido más intensos. Adicionalmente, se comienza a trabajar para que la RN, además de reconocer el modelo más apropiado, haga una pre-estimación de los parámetros de dicho modelo, con lo cual el algoritmo de ajuste final, si fuera aún necesario utilizarlo, sería extremadamente eficiente, pues sus condiciones iniciales de operación serían los valores estimados por la RN. Se usarán patrones para estímulos de entrada tipo rampa y se incluirá en el algoritmo de predicción el modelo matemático que representa la respuesta de los sistemas de primero y segundo orden a estímulos rampa.

REFERENCIAS

- [1] K. Ogata, Ingeniería de control moderna, D.F. México, Prentice Hall, 1998.
- [2] T. Söderström, & P. Stoica, System Identification, Englewood Cliffs, New Jersey, USA, Prentice-Hall, 1989.
- [3] T.F. Edgar & D.M. Himmelblau, Optimization of chemical processes, New York, USA, MacGraw-Hill, 1988.
- [4] D. M. Himmelblau, Applied nonlinear programming, New York, USA, MacGraw-Hill, 1972.
- [5] A.V. Oppenheim & A. Willsky, Señales y sistemas, D.F., México, Prentice-Hall Hispanoamericana, 1998.
- [6] National Instruments, LabVIEW, a graphical programming language to create applications, version 6.1, USA, 2002. www.ni.com/labview.