

A STOCHASTIC HYBRID MODEL IMPLEMENTATION WITH GENETIC ALGORITHMS AND SIMULATED ANNEALING (SA) FOR PARAMETER IDENTIFICATION IN DYNAMIC SYSTEMS

IMPLEMENTACIÓN DE UN MODELO HÍBRIDO ESTOCÁSTICO CON ALGORITMO GENÉTICO Y SIMULATED ANNEALING (SA) PARA LA IDENTIFICACIÓN DE PARÁMETROS EN SISTEMAS DINÁMICOS

MSc(c). Rodrigo Calderón Herreño, PhD. Oscar Eduardo Gualdrón Guerrero
MSc(c). Adrián Carvajal Ferrer

Universidad de Pamplona, Facultad de Ingenierías y Arquitectura.
Ciudadela Universitaria, Pamplona Norte de Santander, Colombia.

E-mail: rodrigo_calderon_h@hotmail.com, oscar.gualdron@unipamplona.edu.co

Abstract: In this work develops an algorithm of highly performance for the lineal identification of systems, utilizing stochastic hybrid methods, such as the combination between the genetic algorithms and simulated annealing. By means of these techniques an estimation of the parameters utilizing is achieved an appropriate structure based on model ARX. The results obtained show an efficient system of identification with a lot greater performance that the conventional media.

Keywords: System identification, stochastic methods, genetic algorithms, simulated annealing, optimization, ARX model.

Resumen: En este trabajo se desarrolla un algoritmo de alto rendimiento para la identificación lineal de sistemas, utilizando métodos híbridos estocásticos, tales como la combinación entre los algoritmos genéticos y *simulated annealing* (recocido simulado). Mediante estas técnicas se logra una estimación de los parámetros utilizando una estructura apropiada basada en modelos ARX. Los resultados obtenidos demuestran un sistema eficaz de identificación con mucho mayor rendimiento que los medios convencionales.

Palabras claves: Identificación de sistemas, métodos estocásticos, algoritmos genéticos, simulated annealing, optimización, modelo ARX.

1. INTRODUCCIÓN

Debido a que los sistemas dinámicos abundan en nuestro medio ambiente, las técnicas de identificación de sistemas han cobrado gran relevancia en diversas áreas del conocimiento (ingenierías, economía, biotecnología, etc.), donde se requiere de un modelo preciso para fines de análisis, predicción, simulación, diseño y control. En particular, las técnicas de control actuales requieren de modelos matemáticos cada vez más exactos para el análisis y el diseño. Siendo que en

muchos casos, tales modelos no pueden ser obtenidos en forma sencilla y económica a partir de las leyes que rigen cada proceso. Aquí es entonces donde juegan un papel decisivo la Identificación de Sistemas Dinámicos, que es una herramienta capaz de proporcionar los métodos necesarios para obtener de manera relativamente sencilla los modelos matemáticos buscados con un alto grado de exactitud. (Altamiranda, *et al.*, 2005; Kunusch, 2003). En términos generales, un *sistema* es un objeto en el cual variables de diferente tipo interactúan y *producen señales observables*.

Las señales observables que nos interesan son llamadas *salidas*. El sistema es afectado por estímulos externos; algunos de los cuales son manipulables por el observador, las *entradas* y por otras, las *perturbaciones*. Estas perturbaciones pueden ser de dos tipos: las medibles directamente y aquellas de las cuales sólo pueden observarse sus efectos en la salida. Para el modelado de sistemas no es muy importante la distinción entre perturbaciones y entradas.

La noción de sistema es, obviamente, muy amplia, por cuanto este concepto representa un papel muy importante en la ciencia moderna. Muchos problemas de diferente índole son solucionados con estructuras orientadas a sistemas. (Vallejo, 1997)

El mayor problema al utilizar algoritmos genéticos en identificación de sistemas, es el gran tiempo de cálculo, debido a que es necesario poblaciones grandes y probar cada individuo de la población en varias muestras, una manera de mejorar el problema anterior es combinar el uso de algoritmos genéticos con simulated annealing (SA), ya que esta última trabaja con pequeñas poblaciones, mejorando así el tiempo de respuesta. De igual manera presenta una sustancial mejoría en la estimación de los parámetros. (Delgado, 1998; Chiamberge, *et al*, 2004; Hoyos, *et al*, 2007).

2. ALGORITMOS GENÉTICOS

Los Algoritmos Genéticos (AG) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza de acuerdo con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin (1859).

Por imitación de este proceso, los Algoritmos Genéticos son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas. Los principios básicos de los Algoritmos Genéticos fueron establecidos por Holland (1975).

Los Algoritmos Genéticos usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los

cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivaldría al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce produciría nuevos individuos (descendientes de los anteriores) los cuales comparten algunas de las características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones.

De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad de que contiene una mayor proporción de buenas características en comparación con la población anterior. Así a lo largo de las generaciones las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda.

3. SIMULATED ANNEALING (SA)

La técnica Simulated Annealing (SA) fue definida en el inicio de la década de los 80, como una nueva herramienta a ser empleada en la solución de grandes problemas combinatoriales. Surgió del campo de la termodinámica como consecuencia de la comparación de problemas formulados en este campo, con los del campo de la investigación operacional; es una metodología simple y de gran potencialidad para ser aplicada a una gran variedad de problemas.

Actualmente, existen algoritmos de búsqueda especializados que consiguen resolver el problema para grandes valores de N.

El Simulated Annealing es una técnica de optimización combinatorial que se usa para afrontar problemas de gran complejidad matemática, de modo que se obtengan soluciones cercanas a la óptima.

La idea original que dio lugar al SA es llamada algoritmo de Metrópolis, el que a su vez está

basado en el método de Montecarlo, con el cual se estudian las propiedades de equilibrio en el análisis del comportamiento microscópico de los cuerpos.

Se fundamenta en el proceso físico de calentamiento de un sólido, seguido por un enfriamiento hasta lograr un estado cristalino con una estructura perfecta. Durante este proceso, la energía libre del sólido es minimizada.

A una temperatura T , en el estado con nivel de energía e_1 , se genera el estado siguiente e_2 a través de una pequeña perturbación. Si la diferencia de energía $e_2 - e_1$ es menor o igual a cero, el estado e_2 es aceptado. Si la diferencia de energía es mayor que cero, el estado e_2 es aceptado con una probabilidad dada por (1), donde KT es la constante de Boltzmann. Si la disminución de la temperatura es hecha de manera paulatina, el sólido puede alcanzar el estado de equilibrio en cada nivel de temperatura.

$$P = e^{\left[\frac{e_1 - e_2}{KT} \right]} \quad (1)$$

El SA aplica la simulación de la secuencia de estados en el proceso de enfriamiento para la solución de problemas de minimización. El costo de la configuración se asocia a la energía y T corresponde al parámetro de control.

A partir del estado e_1 con costo $f(1)$, se genera el estado e_2 con costo $f(2)$. El criterio de aceptación determina si el nuevo estado es aceptado; para esto se calcula la probabilidad.

$$\text{Prob.acep.}e_2 = \begin{cases} 1 & \text{si } f(1) \leq f(2) \\ e^{\left[\frac{e_1 - e_2}{T} \right]} & \text{si } f(1) > f(2) \end{cases} \quad (2)$$

Con esta estrategia se evita el quedar atrapado en mínimos locales. Inicialmente cuando T es grande, se permiten nuevas configuraciones que deterioren la función objetivo. A medida que disminuye la temperatura, la probabilidad de aceptar soluciones peores que la que se tiene es cada vez menor.

Es clave la determinación de la temperatura inicial T , estos parámetros se calibran para adaptarse al tipo y tamaño del problema, de tal manera que se consigan soluciones satisfactoriamente con el SA. (Franco, *et al*, 2005)

4. MODELO ENTRADA-SALIDA (ARX)

Probablemente la relación entrada-salida más simple que se puede obtener sea la proveniente de una descripción del sistema como una ecuación lineal en diferencias.

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + \dots + a_{n_a} y(t-n_a) = b_1 u(t-1) + b_2 u(t-2) + \dots + b_{n_b} u(t-n_b) + e(t) \quad (3)$$

Debido a que el término de ruido blanco $e(t)$ entra como un error directo en la ecuación en diferencias, el modelo (3) es también conocido como *modelo o estructura de ecuación de error*.

En este caso los parámetros a ajustar serán:

$$\mathbf{q} = [a_1 \quad a_2 \quad \dots \quad a_{n_a} \quad b_1 \quad b_2 \quad \dots \quad b_{n_b}] \quad (4)$$

Ahora proponemos dos polinomios $A(q)$ y $B(q)$ de la forma,

$$A(q) = 1 + a_1 q^{-1} + \dots + a_{n_a} q^{-n_a} \quad (5)$$

$$B(q) = b_1 + b_2 q^{-1} + \dots + b_{n_b} q^{-n_b+1}$$

De (3) y (5), tenemos

$$A(q) y(t) = B(q) U(t) + e(t) \quad (6)$$

El flujo de señal será:

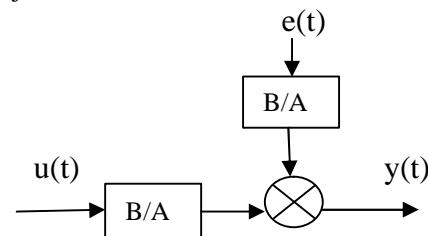


Fig 1 Estructura ARX

A este modelo se lo conoce como estructura "ARX", donde "AR" hace referencia a la parte autorregresiva $A(q).y(t)$ y "X" a la entrada extra (extra input) $B(q).u(t)$ también conocida como variable exógena.

El flujo de señal de la figura 1 nos indica que posiblemente este no sea el modelo más natural desde un punto de vista físico, ya que el ruido blanco es sumado a la salida luego de pasar a través del denominador del sistema dinámico. Sin embargo, el set de modelos de ecuación de error posee una propiedad importante que lo convierte en una buena primera elección en muchas aplicaciones.

5. IDENTIFICACIÓN DE LOS PARÁMETROS

El algoritmo consta de varios pasos figura 2, Primero se genera una población inicial con el algoritmo genético (primera generación), la cual contiene un determinado número de individuos. Cada uno de estos individuos es una solución potencial al problema de optimización. Posteriormente, la población evoluciona (se modifica) mediante la selección de individuos (padres) y su cruce para generar una nueva población (nueva generación de hijos), y a la población resultante se le aplican operadores de mutación (realizado en este caso por *Simulated Annealing*). El método de selección y los operadores de cruce y mutación (S.A) son un punto esencial en el funcionamiento del algoritmo. En cada generación, los individuos son evaluados para ver si resuelven el problema de manera satisfactoria (evaluación de la aptitud). Este proceso es repetido hasta que se obtenga un individuo que logre resolver el problema de optimización dentro de un criterio previamente establecido.

5.1 Selección de individuos

En cada generación de individuos debe hacerse una selección de los individuos más cualificados para pasar sus genes a la siguiente generación, al igual que ocurre con la selección natural. Para acercarse lo más posible a ésta última, la selección en un algoritmo genético se hará de manera aleatoria, pero teniendo cada individuo una probabilidad de ser seleccionado proporcional a su función de *fitness*, es decir, a la bondad de sus genes. De este modo conseguimos que en general se vayan eligiendo a los mejores individuos, pero, al mismo tiempo, los que no son tan buenos también tienen posibilidades de pasar y aportar algún gen que haga falta para formar mejores individuos en la siguiente generación.

5.2 Cruce

Una vez seleccionados los individuos cuyos genes pasarán a la siguiente generación, se procede a la formación de ésta. Para ello primero se realizarán un número aleatorio de cruces.

Los individuos que se vayan a cruzar también se deberán elegir aleatoriamente, teniendo en cuenta que un mismo individuo no puede cruzarse más de una vez. El cruce consiste en un intercambio de

genes entre dos individuos a partir de un punto que, como casi todo, se elige aleatoriamente.

5.3 Condición de parada

Se deben realizar numerosos ciclos de selección, cruce y mutación (en este caso será realizada por el *Simulated Annealing*) para mejorar suficientemente la población de individuos. El algoritmo genético debe finalizar cuando considere que la mejor solución encontrada hasta ese momento es bastante buena o que no va a poder encontrar una mejor. Para eso se define una condición de parada.

La condición de parada más sencilla es hacer que el algoritmo pare siempre después de haber hecho un número de ciclos predeterminado. Otra posible condición de parada es que la función de *fitness* alcance un valor determinado que se considere que es suficientemente bueno para ese problema.

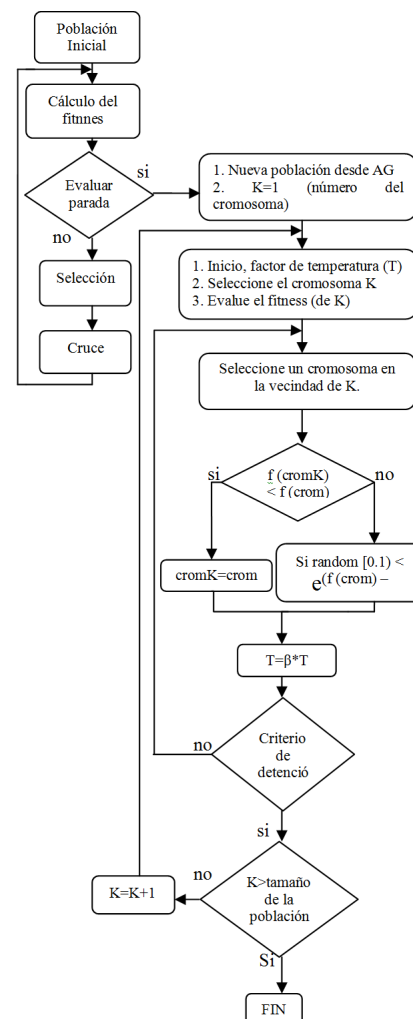


Fig. 2. Diagrama de flujo.

Se guardará el individuo que tenga el valor de *fitness* más alto de la generación inicial. A partir de ahí, si ese valor no es superado en un determinada numero de ciclos, el algoritmo parará. Siempre que el valor de *fitness* del individuo sea superado, se guardará el nuevo mejor individuo y se volverá a contar hasta un número determinado de ciclos del algoritmo.

5.4 Mutación (Simulated Annealing).

El algoritmo parte de la utilización de un conjunto de variables (generalmente todas las variables disponibles, en este caso cromosomas), “cromK”. Una vez calculado el *fitness*, se determina un conjunto “crom” seleccionado en la vecindad de cromK (escogidas aleatoriamente dentro de “cromK”). El *fitness* de “crom” es obtenido y se calcula la diferencia entre el modelo original y el nuevo.

$$f(\text{cromK}) < f(\text{crom})$$

O lo que es lo mismo:

$$?E = \text{Fitness}(\text{nuevo}) - \text{Fitness}(\text{viejo}) \quad (7)$$

Se puede afirmar que ?E es negativo si el nuevo modelo es superior al original, es decir el resultado del cálculo del error de predicción es menor, por lo que la clasificación ha mejorado eliminando dicha variable. Ahora bien, si el modelo resultante al utilizar las variables de “crom” es peor al original (?E positivo) no significa directamente que dicha combinación deba ser rechazada. Se pasaría a una segunda fase en la cual se define la probabilidad:

$$P_i = e^{(f(\text{crom}) - f(\text{cromK})) / T} \quad (8)$$

Donde T es la temperatura de trabajo (se escoge un valor inicial para T). Si $P_i > R$ (R es un valor aleatorio con distribución uniforme entre [0,1]) la nueva solución es retenida y el algoritmo prosigue eliminando variables a partir de “crom”. En caso contrario, el algoritmo prosigue desde “cromK”.

El proceso se ejecuta un número determinado de iteraciones (# iteraciones = N variables totales - 1) para la temperatura T. Finalmente todo lo anterior se repite para $T = \beta T$, ($\beta < 1$). En otras palabras, el proceso de selección se desarrolla a una temperatura más baja (el proceso de aceptación de cambios es menos exigente). El número de temperaturas a computar también es definido a priori. Teniendo en cuenta que un número bajo puede dar resultado a soluciones no muy buenas y un número alto incrementaría drásticamente el tiempo de computación del proceso (un valor

promedio de ejecución puede ser de 50 temperaturas).

6. RESULTADOS DE LA SIMULACIÓN

La data utilizada fue extraída de la base de datos para sistemas de identificación (DAISY), a dicha página se puede tener acceso desde: <http://www.esat.kuleuven.ac.be/sista/daisy/>.

Esta base de datos contiene datos de diferentes procesos, la data seleccionada corresponde a: *Liquid-saturated steam heat exchanger*, el proceso es un intercambiador de calor de líquido-vapor saturado, donde el agua es calentada por vapor saturado a presión a través de un tubo de cobre. La salida variable es la temperatura de salida de líquido. Las variables de entrada son la tasa de flujo de líquido, la temperatura del vapor y la temperatura de entrada del líquido. En esta data la temperatura del vapor y la temperatura de entrada del líquido se mantiene constante a sus valores nominales por lo tanto sólo se tiene una entrada y una salida.

6.1 Validación del modelo

La comparación del modelo se realiza comparando la salida del modelo real versus la del modelo obtenido y por el análisis de residuales figuras 3 y 4. Al observar estas gráficas se puede ver que la respuesta estimada es similar a la respuesta real, generando esto óptimos resultados en la identificación del modelo.

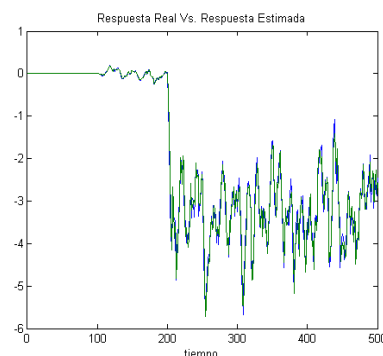


Fig. 3. Salida del sistema y del modelo.

6.2 Modelo obtenido

Los parámetros a_n y b_n del modelo ARX hallados son los siguientes:

$$\begin{aligned} A(q) &= 1 - 1.154 q^{-1} + 0.3085 q^{-2} - 0.0255 q^{-3} \\ B(q) &= -2.296 + 0.04992 q^{-1} - 0.3608 q^{-2} \end{aligned} \quad (9)$$

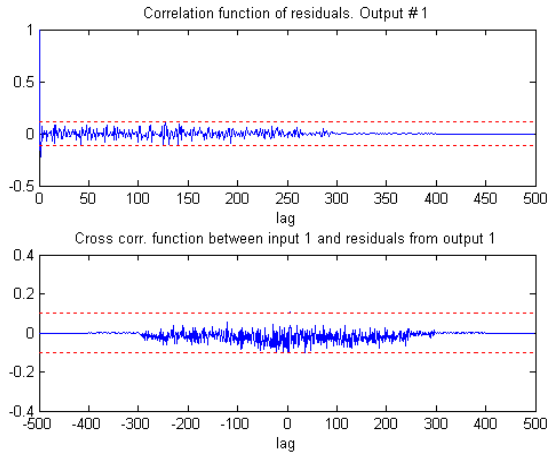


Fig.4. Correlación entre la salida del sistema y del modelo.

Las gráficas y los parámetros fueron halladas utilizando el algoritmo híbrido estocástico con algoritmo genético y *Simulated Annealing* (SA). Este algoritmo se implementó con la herramienta computacional Matlab.

7. CONCLUSIONES

El “sistema real” es una entidad que no puede ser dimensionada totalmente a través de un modelado práctico. En consecuencia, tenemos que conformarnos con descripciones parciales que sean adecuadas para nuestros propósitos. Sin embargo nuestro algoritmo nos genera una muy buena aproximación de la salida estimada con la salida real.

Con técnica del recocido simulado logra mejorar la identificación del sistema reemplazando el operador de mutación en el algoritmo genético, disminuyendo así el tiempo de procesado y generando una buena estimación de los parámetros del modelo utilizado.

A futuro se desarrollara un nuevo algoritmo utilizando diferentes técnicas estocásticas para de esta manera encontrar una nueva técnica de identificación que obtenga respuestas más cercanas a la salida real.

REFERENCIAS

- Edmary Altamiranda, Rodrigo Calderón y Eliezer Colina (2005). *Un Algoritmo Evolutivo Para Identificación de Sistemas Lineales*. Instituto de Investigaciones y Desarrollo de Tecnologías Aplicadas (IIDTA), Universidad de Pamplona, Colombia.
- Cristian Kunusch (2003). “*Identificación De Sistemas Dinámicos*”, Universidad Nacional de la Plata, Facultad de Ingeniería, Departamento de Electrotecnia.
- Eric Vallejo R. (1997). “*Identificación Paramétrica De Sistemas Dinámicos*”, Profesor del Departamento de Ingeniería Eléctrica de la Universidad del Norte. Barranquilla, Colombia.
- Delgado Alberto (1998). “*Inteligencia artificial y minirobots*”, ECOE ediciones, Bogotá, Colombia.
- Chiamberge M., Merelo J. J. *et al.* (2004). “*A Comparacion of neural networks, Linear Controllers, Genetic Algorithms and simulated annealing for real time control*”, Dipartimento di Electronica, Politécnico di Torino, Italia.
- José Gabriel Hoyos G., Jaiber Evelio Cardona, Ramiro Arango. (2007). “*Control En Línea Con Algoritmos Genéticos y Recocido Simulado*”, Scientia et Technica, Agosto No. 35, Universidad Tecnológica de Pereira, Pereira Colombia.
- John Fredy Franco B. Pompilio Tabares E. (2005). “*Aplicación del Simulated Annealing al Problema de las N Reinas*”, Grupo de Planeamiento en Sistemas Eléctricos. Área de Investigación de Operaciones, Scientia et Technica Año Xi, No. 29, Universidad Tecnológica de Pereira, Pereira Colombia.