



Inteligencia Artificial. Revista Iberoamericana
de Inteligencia Artificial

ISSN: 1137-3601

revista@aepia.org

Asociación Española para la Inteligencia
Artificial
España

Irízar, M.; Galán, R.; Jiménez, A.

Algoritmo Genético para la Identificación de Procesos en el Espacio de Estados
Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 4, núm. 10, verano, 2000,
pp. 94-99

Asociación Española para la Inteligencia Artificial
Valencia, España

Disponible en: <http://www.redalyc.org/articulo.oa?id=92541010>

- ▶ Cómo citar el artículo
- ▶ Número completo
- ▶ Más información del artículo
- ▶ Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Algoritmo Genético para la Identificación de Procesos en el Espacio de Estados

M. Irízar(1), R. Galán(2), A. Jiménez(2)

(1) Departamento de Automática y Computación

Instituto Superior Politécnico José Antonio Echeverría, CUBA

e-mail: mirtha@electrica.ispjae.edu.cu

(2) Departamento de Automática, Ingeniería Electrónica e Informática Industrial
Universidad Politécnica de Madrid

e-mail: galan@disam.upm.es

RESUMEN: Este trabajo plantea la aplicación de algoritmos genéticos a la obtención de las ecuaciones de estado y de salida que representan procesos no lineales SISO en los que aparecen términos producto, describiéndose el método y las ventajas de su utilización mediante simulaciones realizadas en MATLAB.

1. INTRODUCCIÓN.

Desde la publicación por Holland de su libro “Adaptación en sistemas naturales y artificiales”, en 1975, un grupo importante de investigadores se ha dedicado al desarrollo y aplicación de los algoritmos genéticos, que se encuentran entre los algoritmos evolutivos más conocidos en las dos últimas décadas. Estos han sido útiles en la solución de variados problemas que requieren la búsqueda en un espacio complejo y no lineal para lograr una solución óptima, ya que su principal ventaja es que no necesitan el control directo del entorno, es decir, no necesitan conocer datos auxiliares sobre el problema que se trata de resolver, y tampoco necesitan recorrer todo el espacio de búsqueda.

Como se conoce, los algoritmos genéticos utilizan un conjunto de cadenas codificadas que representan los parámetros del sistema, las cuales se combinan mediante operadores genéticos (selección, cruce y mutación). Partiendo de una población inicial, se crean generaciones sucesivas a través de la evaluación de una función objetivo. Las cadenas que maximizan la función objetivo en el momento de terminación del algoritmo se toman como soluciones al problema de optimización.

Para obtener mejoras en la ejecución y adaptación de estos algoritmos a diferentes situaciones prácticas se han desarrollado múltiples variantes de operadores.

Muchos problemas de control requieren la identificación del sistema, o sea, la determinación de un modelo adecuado dado un conjunto de datos de entrada/salida, el cual puede ser usado para hacer más efectiva la predicción y el control. Aunque existe una gran variedad de técnicas desarrolladas para esta aplicación, ninguna de ellas ha demostrado ser lo suficientemente efectiva dentro del dominio de casos que se presentan. Muchos de los métodos tradicionalmente usados en la estimación de parámetros suponen la existencia de un óptimo único (o suponen que la solución inicial se encuentra cerca de la solución real), y algunos de ellos no modelan el ruido o requieren un conjunto de datos grande del sistema, los cuales no siempre están disponibles.

En este trabajo se plantea la aplicación de algoritmos genéticos a la identificación de sistemas SISO (una entrada, una salida) no lineales, considerándose como posibles la existencia de términos producto en la ecuación de estado, característica presente en muchos procesos químicos, demostrándose las ventajas de su uso a través de simulaciones.

2. IDENTIFICACIÓN DE SISTEMAS EN EL ESPACIO DE ESTADOS.

En la identificación de sistemas constituyen aspectos importantes la determinación de la estructura del modelo y su validación. La identificación de sistemas fuera de línea es un método batch que procesa los datos almacenados “simultáneamente” [4]. En la misma se supone conocida la estructura del modelo y se estiman los parámetros del mismo utilizando algún método numérico. El problema de la estimación de parámetros es un problema generalmente multimodal, o sea, tiene múltiples óptimos o soluciones.

Se han desarrollado trabajos previos relacionados con la identificación, que establecen los siguientes pasos para la estimación de parámetros de un modelo mediante un algoritmo genético [1]:

1. Elegir y construir el modelo del sistema a identificar.
2. Seleccionar la función o índice que indicará la fuerza de cada posible solución.
3. Elegir una regla para codificar, en un cromosoma, los parámetros a estimar.
4. Determinar los operadores genéticos que se utilizarán, y si es necesario desarrollar aquellos que se necesiten.
5. Determinar los valores para los parámetros que controlan el AG.
6. Hacer los experimentos necesarios.

Con la disponibilidad de las computadoras digitales, resulta conveniente formular las ecuaciones que representan los sistemas de control en el dominio del tiempo, pues dichas técnicas pueden ser utilizadas para sistemas no lineales, variantes en el tiempo y multivariados [6], constituyendo una base esencial para la teoría del control moderna y la optimización de sistemas.

Las variables de estado describen la respuesta futura de un sistema, dado el estado presente, las entradas de excitación y las ecuaciones que describen la dinámica. Estas no son un conjunto único, pudiendo seleccionarse conjuntos alternativos de variables de estado.

Evaluación en tiempo discreto de la respuesta del sistema.

La respuesta de un sistema representado por una ecuación diferencial del vector de estados puede ser obtenida utilizando una aproximación en tiempo

discreto, método útil para evaluar la respuesta en el tiempo de sistemas no lineales.

La ecuación diferencial del vector de estado básica puede ser escrita como:

$$\dot{x} = f(x, u, t)$$

donde f es una función, no necesariamente lineal, del vector de estado y del vector de entrada u .

Usando la ecuación de la aproximación de la derivada y resolviendo para

$x(k+1)$ cuando $t=kT$, se obtiene:

$$x(k+1) = x(k) + T[f(x(k), u(k), k)]$$

Entonces las ecuaciones diferenciales de primer orden para un sistema con posibles términos no lineales producto y la ecuación de salida, pueden ser escritas en forma general como se muestra en la Fig. 1.

Para demostrar el funcionamiento del algoritmo genético se utilizará un sistema con las ecuaciones de estado y de salida expresadas en notación matricial, según se muestra en la Fig. 2, donde las matrices

$$M_{1,2} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

representan los coeficientes de los términos no lineales presentes en la ecuación de estado.

A continuación se explicará la forma de realización de los diferentes pasos.

Función de evaluación.

El índice del grado de validez de un modelo está dado por el error cuadrático entre las salidas reales y las salidas propuestas por el mismo; este error es calculado mediante la ecuación:

$$e = \sum_{i=0}^m (y_i - y_m)^2$$

donde m es el número de datos utilizados. El mejor modelo es el que tenga el menor error, por tanto el AG deberá minimizar esta ecuación.

Método de codificación.

Para el AG cada cromosoma representa una solución al problema planteado y, por tanto, un posible conjunto de coeficientes para las ecuaciones de estado y salida elegidas.

$$x_i(k+1) = x_i(k) + T \{ [x_1(k)x_2(k)...x_n(k)u_1(k)u_2(k)...u_m(k)]^* M_i * \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \\ u_1(k) \\ u_2(k) \\ \vdots \\ u_m(k) \\ 1 \end{bmatrix} \}$$

$$y(k) = [c_1 c_2 ... c_n] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix}$$

Fig. 1. Ecuaciones de estado y de salida de un sistema con términos no lineales producto en tiempo discreto.

$$x_1(k+1) = x_1(k) + T \{ [a_{11}x_1(k) + a_{12}x_2(k) + b_1u(k)] + [x_1(k)x_2(k)u(k)]^* M_1 * \begin{bmatrix} x_1(k) \\ x_2(k) \\ u(k) \end{bmatrix} \}$$

$$x_2(k+1) = x_2(k) + T \{ [a_{21}x_1(k) + a_{22}x_2(k) + b_2u(k)] + [x_1(k)x_2(k)u(k)]^* M_2 * \begin{bmatrix} x_1(k) \\ x_2(k) \\ u(k) \end{bmatrix} \}$$

$$y(k) = [c_1 c_2] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

Fig. 2. Ecuaciones de estado y de salida del sistema ejemplo.

Cada coeficiente se codifica como un valor real comprendido entre -10 y $+10$, cuyos valores iniciales se generan aleatoriamente, y cuyo conjunto constituye la población inicial para el algoritmo. Los intervalos seleccionados para los parámetros

corresponden a criterios prácticos. En los casos donde se conozca que los parámetros están fuera de este intervalo, se puede modificar el software con cambios mínimos.

Cada individuo contiene una posible solución (juego de coeficientes) para los parámetros del modelo propuesto y cada coeficiente se encadena con los otros, formando un cromosoma de 26 bits de longitud. Esta codificación se muestra en la Fig. 3.

$$\begin{aligned} &a_{11}a_{12}a_{21}a_{22}b_1b_2m1_{11}m1_{12}m1_{13}m1_{21}m1_{22}m1_{23} \\ &m1_{31}m1_{32}m1_{33}m2_{11}m2_{12}m2_{13}m2_{21}m2_{22}m2_{23} \\ &m2_{31}m2_{32}m2_{33}c_1c_2 \end{aligned}$$

Fig. 3. Esquema de codificación.

Operadores Genéticos.

El AG utilizado para la identificación incluye los tres operadores básicos: reproducción, cruce y mutación. En la reproducción mediante torneo estocástico se realizan torneos entre individuos vecinos y para poder lograr la minimización, se elige como ganador a aquel que tiene la menor fuerza.

Al mejor individuo se le asignan dos copias en la generación siguiente y al peor, ninguna

El cruce consiste en hallar el valor promedio de dos individuos cuyas aleatoriamente en la población, de acuerdo a la probabilidad establecida.

La mutación consiste en alterar los valores de los coeficientes generados

por un individuo en una cantidad dada. El algoritmo para la mutación es el siguiente:

Para todos los individuos de la población,

Para cada parámetro del cromosoma,

Generar valor aleatorio entre 0 y 1

(valeat1)

Si valeat1<= Probab. de mutac.

Generar valor aleatorio entre 0 y 1

(valeat2)

Si valeat2<=0.5

Parám.=(0.5)*Parám

Si valeat2>0.5

Parám.=(-0.5)*Parám

Fin

A continuación se resume el algoritmo genético desarrollado:

- Entrar parámetros del algoritmo, error máximo del modelo e intervalo de muestreo.

- cont=0;

- Para el número de generaciones,

- Evaluar cada individuo de la población de acuerdo a la función de fuerza y determinar el mejor

- Generar aleatoriamente las posiciones para el torneo y efectuarlo

- Generar aleatoriamente las posiciones para el cruce y efectuarlo según probabilidad

- Realizar la mutación según probabilidad
- Introducir mejor individuo en la población(elitismo)
- Calcular error
- Si se mantiene el mismo error

- Incrementar cont

- Si el contador llega a 10,
o si el error es menor que
el máximo:

- Terminar ciclo

- Mostrar gráficos de salida y matrices obtenidas para el mejor individuo

Fin

Como se observa, la ejecución del algoritmo puede terminar porque se haya alcanzado un valor mínimo de error, porque se mantenga dicho valor constante durante 10 generaciones de forma consecutiva o porque se llegue al número de generaciones establecidas previamente.

3. RESULTADOS.

Los experimentos que se describen tienen como objetivo validar el software siguiendo dos criterios fundamentales:

- Es capaz de optimizar una función usando los operadores del algoritmo genético, incrementando la idoneidad de la población a través de las generaciones.
- Es capaz de identificar correctamente un proceso, mostrando la utilidad de los algoritmos genéticos en este campo.

Para probarlo, se reflejan los ejemplos más significativos. En cada caso se muestra un gráfico con el valor del error a través de las generaciones y, para validar la identificación, uno con la salida real y la salida del modelo estimado.

Los dos primeros experimentos tienen como objetivo analizar el desempeño del algoritmo genético con la variación de la señal de entrada. En ambos experimentos el número de generaciones, el tamaño de la población, la probabilidad de cruce y la de mutación se mantuvieron constantes, y los datos que se utilizaron corresponden a un sistema descrito por las siguientes ecuaciones:

$$x_1^1 = -x_1 - u + x_2 u$$

$$x_2^1 = -x_2 + u$$

$$y = x_1 + x_2$$

Experimento 1.

Este experimento se realizó con una población de 300, una probabilidad de cruce de 0.8 y una

probabilidad de mutación de 0.03, aplicándose una señal de entrada aleatoria. (Ver Fig. 4).

El error en la estimación de los coeficientes de las ecuaciones de estado y salida del proceso fue decreciendo, estabilizándose finalmente en un valor de 5.49.

Experimento 2.

Este experimento se realizó con los mismos parámetros del algoritmo que en el caso anterior, pero en este caso se usó una señal de entrada sinusoidal.

El error final en este experimento fue 0.02 (Fig. 5).

El segundo bloque de experimentos tuvo por objetivo analizar el comportamiento del algoritmo genético con diferentes sistemas. En los mismos se mantuvo constante el número de generaciones (100), el tamaño de la población (300), la probabilidad de cruce (0.8) y la probabilidad de mutación (0.03). Los experimentos se llevaron a cabo partiendo de sistemas con diferentes tipos de términos no lineales, los que se especifican en cada caso.

Experimento 3.

En este experimento (Ver Fig. 6) los datos que se utilizan corresponden a un sistema descrito por las siguientes ecuaciones:

$$\begin{aligned}x_1^1 &= -x_1 - u - x_1 x_2 + x_2 u \\x_2^1 &= -x_2 + u + x_1^2 \\y &= x_1 + x_2\end{aligned}$$

Experimento 4.

En este experimento (Fig. 7) los datos corresponden a un sistema descrito por las ecuaciones:

$$\begin{aligned}x_1^1 &= -x_1 - u - x_1 x_2 \\x_2^1 &= -x_2 + u - x_1^2 \\y &= x_1 + x_2\end{aligned}$$

Como se observa en ambos casos, el algoritmo responde de forma muy similar al sistema real ante una señal de entrada aleatoria.

Debe destacarse que en el algoritmo genético se parte de un conjunto de datos de entrada y salida simulados para la ejecución de los diferentes experimentos, lo que en la práctica correspondería al estímulo de la planta y la adquisición de valores representativos en diferentes puntos de funcionamiento, existiendo la posibilidad de

aumentar o disminuir dichos datos de acuerdo a las disponibilidades del sistema.

4. CONCLUSIONES.

El software utilizado para llevar a cabo los experimentos fue desarrollado en MATLAB.

Los resultados obtenidos demuestran que los algoritmos genéticos son aplicables a la estimación de parámetros trabajando en el espacio de estados para sistemas lineales y algunos tipos de sistemas no lineales, constituyendo una herramienta simple, pero pueden existir otras alinealidades tales como funciones seno, coseno y raíces cuadradas, que también deben ser consideradas en el algoritmo, lo que sugiere la necesidad de continuar explorando nuevas alternativas para su aplicación a sistemas no lineales con diferentes características.

5. REFERENCIAS.

- [1] Romero, J.L y Valenzuela,M.(1992). Aportaciones de los AGs a la Identificación paramétrica de sistemas. Memorias de Informática'92.
- [2] Marín, F., García, F., y Sandoval, F.(1992). Algoritmos genéticos: una estrategia para la búsqueda y optimización. Informática y Automática, vol 26-3.
- [3] Goldberg, D.E.(1985).Genetic Algorithms in search, optimization, and machine learning. M.A. Addison- Wesley.
- [4] Söderstrom, T. y Stoica, P.(1989). System Identification. Prentice Hall International.
- [5] Man, K. F., Tang, K.S., y Kwong, S. (1996). Genetic Algorithms: Concepts and Applications. IEEE Transactions Industrial Electronics, vol. 43, No. 5.
- [6] Dorf, R.C. y Bishop, R. H. (1998). Modern Control Systems. Addison Wesley Longman, Inc.

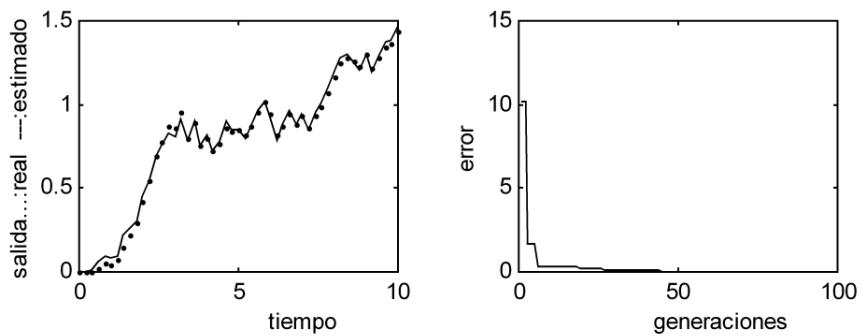


Fig . 4 Resultados del experimento 1.

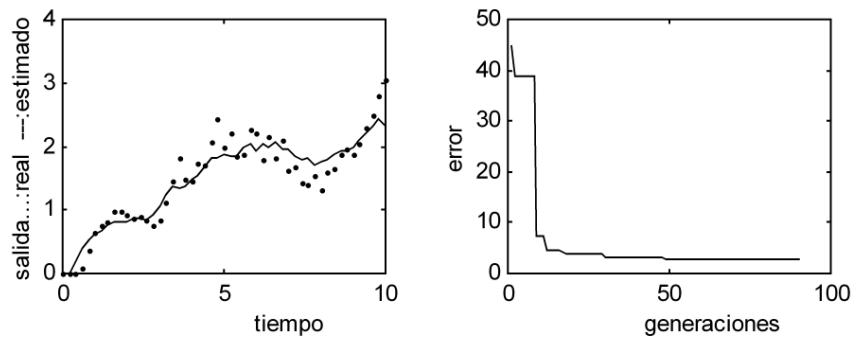


Fig. 5 Resultados del experimento 2.

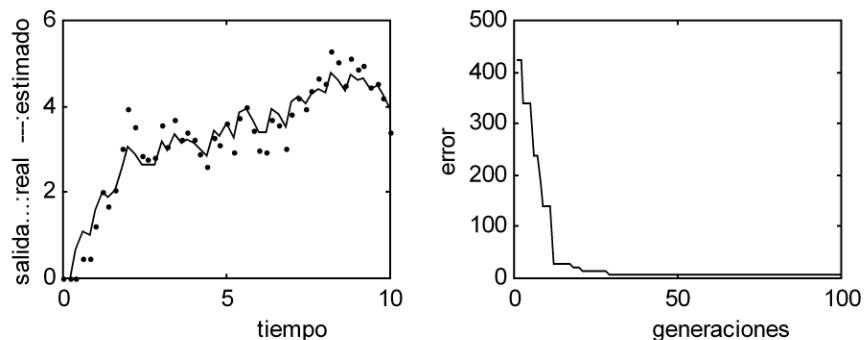


Fig. 6 Resultados del experimento 3.

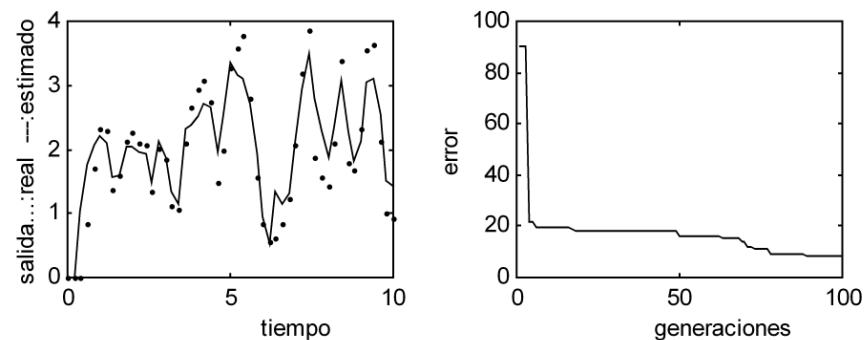


Fig. 7 Resultados del experimento 4.