

Búsqueda SA-P de Estructuras y Parámetros de Modelos de Procesos Dinámicos

A. López¹, L. Sánchez²
Universidad de Oviedo.

¹ Departamento de Ingeniería Eléctrica. C. Universitario de Viesques.
Ed. Departamental N.º.2, 33204, Gijón, Asturias.
e-mail: antonio@isa.uniovi.es

² Departamento de Informática. C. Universitario de Viesques.
Ed. Departamental N.º.1, 33204, Gijón, Asturias.

e-mail: luciano@lsi.uniovi.es

Resumen—

Los enfoques más eficientes para la identificación evolutiva de procesos dinámicos se basan en prueba-error siguiendo un enfoque jerárquico: en primer lugar se propone una nueva estructura y a continuación se determina el conjunto de parámetros numéricos, repitiéndose el proceso hasta que se encuentra un modelo lo suficientemente preciso. La búsqueda se ha realizado siguiendo diferentes esquemas, habitualmente en base a la Programación Genética, yendo desde la aplicación del GP canónico, pasando por la aplicación de métodos de aprendizaje jerárquicos donde se realiza una búsqueda GP de la estructura y se aplica un método determinista de optimización numérica para la identificación paramétrica y llegando hasta métodos donde se realiza una búsqueda evolutiva concurrente en el espacio de estructuras y parámetros mediante el algoritmo GA-P, los cuales se han mostrado comparativamente más eficientes que los anteriores.

En este trabajo, se compara el algoritmo GA-P aplicado al modelado de procesos dinámicos con un algoritmo de optimización estocástico de reciente aparición como es el SA-P. Se aplica a problemas sintéticos y reales. Los resultados parecen ser favorables al algoritmo SA-P.

Palabras clave— Algoritmos Genéticos, Programación Genética, GA-P, SA-P, Sistemas Dinámicos, Modelado, Identificación.

I. INTRODUCCIÓN

Gran parte de las aplicaciones de métodos evolutivos a la identificación de procesos se centran en la búsqueda de modelos no lineales en espacio de estados. El objetivo del proceso de aprendizaje es la obtención de un conjunto de ecuaciones que definan la dinámica del proceso. Desafortunadamente, en la práctica (a no ser en la resolución de problemas de pequeño tamaño) es difícil obtener un conjunto de ecuaciones que relacionen todas las variables de estado consideradas. Aparte de este problema, otra desventaja de los modelos en espacio de estados es que en general se prefieren representaciones modulares, las cuales permiten determinar grupos de variables afectados por un

conjunto de parámetros específico.

La Programación Genética ha sido aplicada con bastante asiduidad al aprendizaje de modelos de procesos utilizando tales representaciones modulares. En un punto posterior se analizarán más detenidamente los esquemas más interesantes. Baste decir por el momento que todos los esquemas citados, de los cuales el más eficiente es el GA-P, se basan en la evolución de una población de modelos mediante la aplicación de distintas técnicas al ajuste estructural y al ajuste paramétrico. No obstante, existen en el estado del arte diferentes alternativas para la optimización de funciones de tal naturaleza. Recientemente, se ha desarrollado un nuevo algoritmo estocástico de optimización, el SA-P [18], como una variante del algoritmo de recocido simulado adaptada para la optimización estructural y paramétrica de un individuo sometido a evolución. Partiendo de un individuo inicial, se aplican mutaciones a la componente estructural o paramétrica según probabilidades definidas. La aceptación o rechazo de un nuevo individuo como sustituto del actual depende de las aptitudes de ambos y de la política de enfriamiento.

A priori, este algoritmo parece una alternativa interesante en cuanto a esquema de optimización. Quizá dos de sus grandes ventajas son su bajo consumo de memoria debido a que maneja una población de un solo individuo (en contra de los enfoques anteriores donde los tamaños de población son de varios centenares) y a que la propia estructura de búsqueda, basada únicamente en el operador de mutación, puede evitar algunos problemas del método evolutivo como el estancamiento prematuro, el cual era aliviado en alguno de los esquemas propuestos [1] mediante técnicas de nichos.

El objetivo de este trabajo es mostrar resultados iniciales comparativos entre las estrategias

de búsqueda GA-P y SA-P, ambas aplicadas a problemas que, si bien son de tamaño reducido, pueden permitir estimar el potencial de las estrategias de búsqueda.

II. ESTRUCTURA DEL ARTÍCULO

Una vez realizada la introducción, la sección III hace referencia a las aplicaciones más destacables de algoritmos evolutivos al modelado de procesos dinámicos, haciendo énfasis en la selección del método GA-P como uno de los que mejores resultados proporciona. Posteriormente, en la sección IV se describen brevemente los esquemas utilizados en este trabajo para pasar a continuación a realizar su comparación (sección V) en el modelado de diferentes procesos. El trabajo finaliza con las conclusiones obtenidas y las posibles líneas de trabajo futuro.

III. PROGRAMACIÓN GENÉTICA EN EL MODELADO DE SISTEMAS DINÁMICOS

Uno de los primeros ejemplos de aplicación se puede examinar en [9], donde se utiliza un Algoritmo Genético estructurado con una representación arbórea (en la práctica, un programa genético). El conjunto funcional propuesto está compuesto de funciones cuadráticas de dos entradas. No obstante, esta representación no resulta de gran utilidad en el modelado de sistemas dinámicos de cara a su uso en el mundo del control de procesos. No es una representación familiar a un ingeniero de control, por lo que las soluciones no le aportan conocimiento en su dominio de trabajo cotidiano. Algunas aplicaciones más cercanas a prácticas usuales se pueden encontrar en [3], [4], [5], [6], [10], [17], [19] y otros enfoques, aunque poco comunes, para modelar sistemas dinámicos se pueden encontrar en [7], [15], [20]. Quizá, como último punto destacable de las aplicaciones citadas, comentar que una característica común a la mayoría de estos esquemas es que incorporan el concepto de dinámica mediante la extensión del conjunto de variables de entrada con variables de entrada/salida retrasadas en el tiempo, tal y como se hace habitualmente en el aprendizaje de modelos dinámicos con métodos especialmente diseñados para la obtención de modelos estáticos (como podría ser, sin ir más lejos, el método de mínimos cuadrados, profusamente utilizado para el ajuste paramétrico de los coeficientes de una ecuación en diferencias en la identificación de procesos dinámicos).

En contraposición a los métodos anteriores, un enfoque muy interesante dada su capacidad representativa en un dominio lingüístico cercano al ingeniero de control, es la búsqueda de un modelo

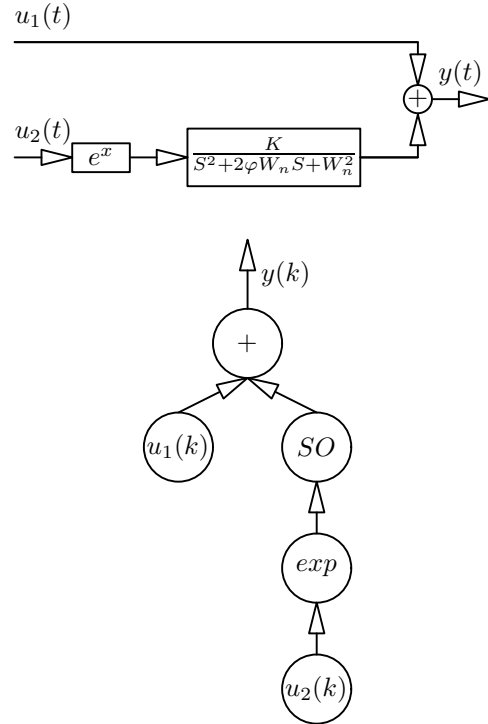


Fig. 1. Diagrama de bloques (parte superior) y representación arbórea (parte inferior) utilizada en algunos esquemas de Programación Genética que buscan un modelo del sistema objetivo en representación de diagramas de bloques.

del proceso objetivo en una representación basada en diagrama de bloques, bien utilizando una representación arbórea [11], [13], [14] (ver figura 1) o basada en grafo [1], [12] (ver figura 2). El conjunto funcional incluye elementos dinámicos definidos en las variables complejas S o Z , haciendo intrínsecas las características dinámicas a la búsqueda, evitando de esta manera artificios como los utilizados en otros enfoques.

Dentro de la Programación Genética aplicada al modelado de sistemas dinámicos, es quizá en los enfoques con individuos basados en diagramas de bloques donde más esfuerzo se ha volcado, lo cual se ha traducido en la prueba de diferentes esquemas de búsqueda.

El primer esquema ensayado consistió en la búsqueda de modelos [13], [14] mediante un enfoque jerárquico, aplicando un GP a la búsqueda estructural y un método de optimización determinista (Hooke-Jeeves) a la búsqueda de unos valores óptimos para sus parámetros. Cada modelo considerado se somete a un ajuste paramétrico mediante la aplicación del algoritmo de optimización. Los operadores genéticos definidos afectan solamente a la componente estructural de los modelos.

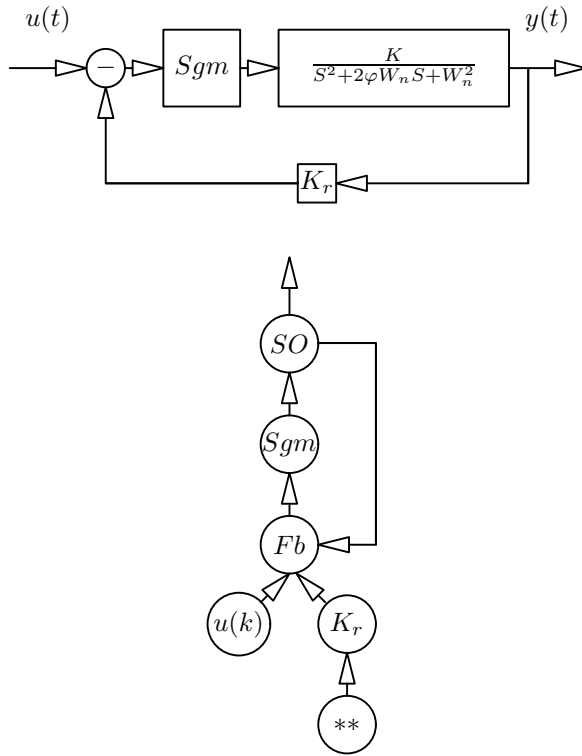


Fig. 2. Diagrama de bloques (parte superior) y representación genérica basada en grafos (parte inferior) que alivia alguno de los problemas de la representación arbórea.

Experimentaciones posteriores [12] mostraron que la propia identificación paramétrica de los modelos manejados es un problema de carácter multimodal, haciendo que la aplicación de un método determinista falle en ocasiones, lo que llevó a considerar la sustitución del método determinista por un método evolutivo, en concreto un algoritmo genético, obteniéndose mejores resultados al no estancarse la búsqueda en mínimos locales estructurales.

No obstante, un algoritmo genético aplicado de forma monolítica sobre cada uno de las estructuras inducidas por el GP supone un gran esfuerzo computacional, sobre todo en lo referente al trabajo realizado en la optimización paramétrica de estructuras no válidas. Por ello se buscó una estrategia que permitiese relajar el esfuerzo computacional en la identificación paramétrica de estructuras, permitiendo realizar tal proceso de una forma progresiva en paralelo a la evolución estructural. Ello llevó a la definición de un algoritmo que sigue una estrategia GA-P [8], y que por tanto permite realizar una búsqueda evolutiva concurrente en el espacio de estructuras y parámetros, mejorando la eficiencia de los esquemas anteriores [1], [2] ante un esfuerzo computacional

similar.

IV. ESQUEMA COMPARADOS

A continuación se describen brevemente las características de los dos métodos de búsqueda comparados en este trabajo comentando sus particularidades de aplicación.

Para ambos esquemas, se ha utilizado una representación basada en grafos dada la imposibilidad de representar algunas estructuras comunes en diagramas de bloques mediante árboles. Los grafos estarán compuesto por nodos funcionales de carácter lineal estático (ganancia estática, sumas, restas, etc.), de carácter lineal dinámico (sistemas de primer orden, segundo orden subamortiguado, etc.) y de algunos elementos de carácter no lineal estático comunes en procesos físicos como podrían ser zonas muertas o saturaciones.

A. Algoritmo GA-P para el modelado de procesos dinámicos

El algoritmo GA-P propuesto se basa en la evolución de una población de modelos en base a operadores de reproducción, búsqueda estructural (cruce a nivel de subárbol y mutación a nivel de subárbol, a nivel de nodo y un operador especial para la mutación de realimentaciones) y operadores de búsqueda paramétrica (cruce y mutación paramétrica junto con un operador de mutación especial que realiza una búsqueda en base al método Simplex de Nelder & Mead [16]) según unas probabilidades predefinidas. Las operaciones de modificación estructural afectan únicamente a la componentes estructural de los individuos involucrados y las de búsqueda paramétrica a sus vectores de coeficientes numéricos.

Para evitar la convergencia prematura del método, se aplica una estrategia de nichos.

B. El Algoritmo SA-P

La evolución del algoritmo SA-P se basa en la modificación del individuo sometido a evolución mediante un operador de *macromutación* (ver Fig. 3). Este operador distingue su ámbito de aplicación, decidiendo si se debe mutar la componente estructural o paramétrica en función de un parámetro de diseño del algoritmo p . Asimismo, maneja dos factores, K_1 y K_2 , que de alguna forma indican el margen de variación admitido para la mutación.

En base a ese único operador, el algoritmo SA-P evoluciona según los criterios clásicos del recocido simulado (ver figura 4). Depende de una temperatura inicial, especificada como parámetro, y de una temperatura final o de un número máximo de evaluaciones de la función objetivo,

```

algoritmo macromutacion es
necesita: C, p, K_1, K_2
produce: C1
si U(0,1)<p entonces
  A = vector aleatorio parámetros reales
  parámetros(C)=
    parámetros(C)*(T/K_1)+(1-(T/K_1))*A
sino
  repetir
    E=expresión(C)
    Seleccionar subárbol(C)
    Reemplazar subárbol seleccionado
      por uno generado aleatoriamente
  hasta que
    DistanciaEdición(E, expresion(C))<T/K_2
  expresion(C)=E
fsi

```

Fig. 3. Operador de macromutación del algoritmo SA-P. La modificación afecta a la componente estructural o paramétrica del individuo sometido a evolución en función de una probabilidad predefinida p .

```

algoritmo SA-P es
necesita: factor enfriamiento, Tinicial, Tfinal,
  p, K_1, K_2, evaluaciones_maximas
produce: CMejor

T=Tinicial CMejor=C=individuo aleatorio
mientras T>Tfinal y
  evaluaciones<evaluaciones_maximas
  C1=macromutacion(C,p,T,K_1,K_2)
  delta = error(C1)-error(C)
  v=valor aleatorio segun U(0,1)
  si (delta<0) o (v<exp(-delta/T))
    C=C1
    si error(C)<error(CMejor)
      entonces CMejor = C
  fsi
  T=T*factor enfriamiento
fmientras

```

Fig. 4. Algoritmo SA-P. Se aplican los métodos de búsqueda habituales del Recocido Simulado.

que será utilizado para la detención del proceso. La temperatura en cada iteración se modificará según la tasa de enfriamiento especificada. Asimismo, la aceptación de un nuevo individuo como sustituto del actual dependerá de los errores de ambos individuos y de la temperatura del algoritmo en la iteración en curso.

En lo referente a la amplitud de variación efectuada por la operación de mutación, se observa que esta se restringe tanto en la componente estructural como paramétrica según una función de la temperatura actual y de dos parámetros de diseño. En cuanto a la variación estructural, se requiere la definición de una función para el cálculo de la distancia de edición. No obstante, la definición de tal función no está clara en nuestro caso de aplicación, por lo que no se utilizará tal restricción, aceptándose como resultado de la mutación el primer intento. Asimismo, experimentaciones

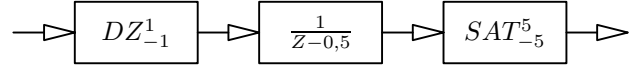


Fig. 5. Proceso sintético utilizado para la comparación de los esquemas analizados.

previas nos han llevado a concluir que el algoritmo evoluciona mejor cuando no se restringe la variación del vector de parámetros según lo indicado en el procedimiento base, por lo que tampoco se aplicará tal técnica. La mutación paramétrica de un individuo será el resultado de variar su vector de parámetros según un valor aleatorio en cada dimensión sin ninguna restricción.

V. RESULTADOS NUMÉRICOS

Para la comparación de ambos esquemas se han seleccionado dos problemas de complejidad baja, a fin de realizar un estudio inicial de ambas estrategias. Uno de ellos consiste en un problema sintético, donde el modelo objetivo está formado por una asociación serie de diferentes componentes de naturaleza lineal y no lineal. El otro problema consiste en la obtención de un modelo de la velocidad de un motor de corriente continua en bucle abierto.

Para ambos casos se mostrará el resultado de diez experimentos independientes de la aplicación del esquema GA-P y del esquema SA-P. El criterio de comparación será el análisis de los resultados de ambos aplicados bajo un esfuerzo computacional similar (permitiendo un mismo número de evaluaciones de la función objetivo).

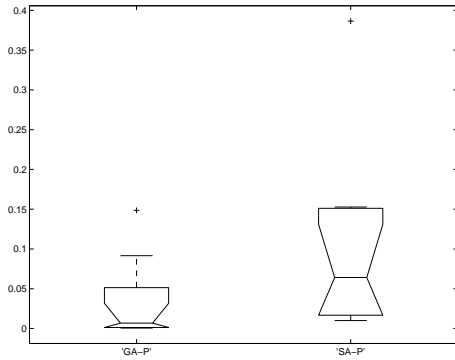
A. Problema Sintético

El modelo objetivo está formado por una asociación serie de una zona muerta, un componente lineal de dinámica de primer orden y una saturación (ver figura 5). Experimentaciones anteriores han mostrado que para este tipo de sistemas una búsqueda evolutiva jerárquica en el espacio de estructuras y parámetros proporciona buenos resultados, mientras que la hibridación de una búsqueda evolutiva en el espacio de estructuras con un método determinista para el ajuste de parámetros conduce al estancamiento de la optimización en mínimos locales estructurales, impidiendo así el hallazgo del mínimo global de la función objetivo.

La Fig. 6 muestra los errores obtenidos en la aplicación de ambos esquemas al modelado del proceso sintético. Previamente al análisis comparativo, es necesario realizar algunas observaciones. En principio, los resultados se pueden catalogar en dos grandes grupos: aquellos en los cuales el error es superior a 0.1 y aquellos en los cuales

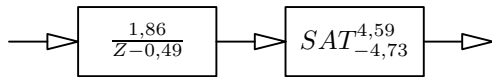
Experimento	GA-P	SA-P
a	0.0104	0.152755
b	0.0005	0.052048
c	0.0012	0.028809
d	0.0038	0.151154
e	0.0005	0.148307
f	0.0053	0.014007
g	0.1485	0.386498
h	0.0514	0.016677
i	0.0082	0.076174
j	0.0915	0.009950
Average	0.0321	0.0884

(a) Errores Numéricos.

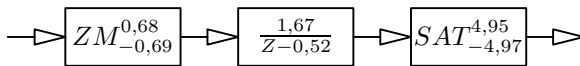


(b) Gráfico de Cajas.

Fig. 6. Resultados numéricos del modelado del sistema de la figura 5. Se han resaltado aquellos experimentos en los que la búsqueda ha convergido a un mínimo estructural.



(a) Experimento e).



(b) Experimento i).

Fig. 7. Resultados de dos experimentos realizados con el algoritmo SA-P en el modelado del sistema de la figura 5.

está por debajo. La razón de esta división es que comprobando los modelos resultado, se observa que en aquellos casos en los que se ha encontrado la estructura del modelo objetivo, el error está por debajo de esa cota, mientras que cuando la estructura del modelo objetivo no es encontrada el error la supera. Por razones de espacio, no se incluyen en este artículo los resultados hallados en todos los experimentos. No obstante, a modo de muestra se pueden observar los mejores individuos hallados por el algoritmo SA-P en los experimentos e) e i) (ver Fig. 7).

Una vez efectuadas las consideraciones anteriores, se puede comprobar que los resultados muestran una cierta ventaja del esquema de búsqueda GA-P respecto al SA-P. El primero es capaz de encontrar la verdadera estructura del modelo en 9 de los diez experimentos realizados, con la consiguiente reducción del error de validación que se traduce en un error promediado de la experimentación bajo. Por su parte, el SA-P es capaz de encontrarla en 6 de los diez experimentos, por lo que el estancamiento del método en los restantes cuatro experimentos ocasiona el aumento del error promediado. No obstante, se ha realizado un diagrama de cuadros para analizar si la diferencia de los valores promediados es significativa, comprobándose que no se puede afirmar que ninguno de los algoritmos sea realmente superior al otro.

B. Problema Real

El problema abordado consiste en la obtención de un modelo de la velocidad de un motor de corriente continua en bucle abierto (ver figura 8). Este tipo de procesos suele modelarse mediante un comportamiento de dinámica de primer orden. En modelos más precisos, el orden de la dinámica suele elevarse a dos, aunque el efecto del segundo polo suele despreciarse por ser poco significativo. Asimismo, suelen presentar una zona muerta como consecuencia de la inercia inicial que es necesario vencer para pasar del estado de reposo al estado de funcionamiento.

La experimentación realizada se ha basado en muestras obtenidas del proceso mediante señales de excitación de baja tensión, con el objetivo de hacer visible la zona muerta. La figura 9 incluye los resultados obtenidos mediante la aplicación de los esquemas GA-P y SA-P.

Se puede observar una vez más cómo se obtienen resultados similares. Si bien en términos promediados sobre los diez experimentos el algoritmo GA-P parece ligeramente más eficiente, ello se debe al experimento g) del SA-P, donde se obtiene un error muy alto (convergencia a un

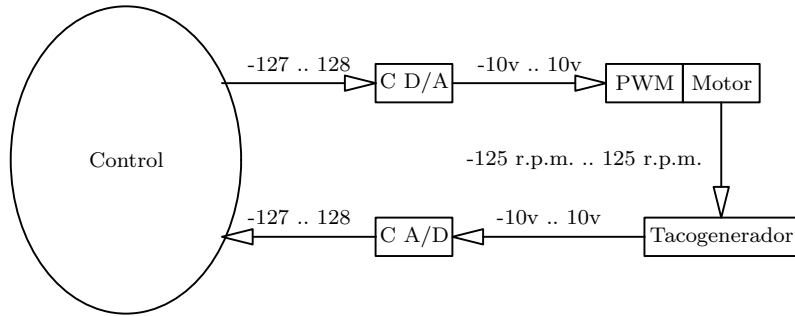
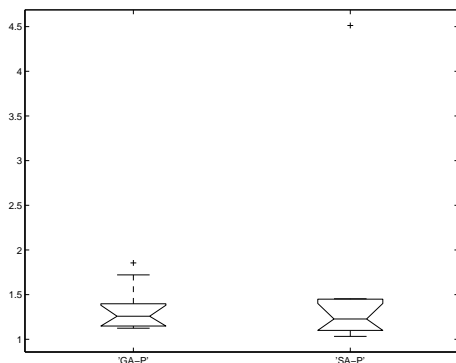


Fig. 8. Proceso real utilizado para la comparación de los esquemas analizados: Motor de corriente continua en bucle abierto.

mínimo local). Ignorando este experimento (eliminando por ejemplo el peor resultado obtenido por los dos algoritmos) se aprecia cómo el SA-P supera el rendimiento del GA-P. Una vez más, el análisis de un diagrama de cuadros de los experimentos muestra que no se puede afirmar que un método sea más eficiente que el otro.

Experiment	GA-P	SA-P
a	1.721107	1.230861
b	1.220925	1.032117
c	1.124030	1.093276
d	1.149357	1.441063
e	1.148093	1.453996
f	1.397652	1.224213
g	1.303401	4.512724
h	1.294362	1.448788
i	1.147530	1.099917
j	1.855206	1.216553
Average	1.3362	1.5754

(a) Errores Numéricos



(b) Gráfico de Cajas

Fig. 9. Resultados numéricos del modelado del motor de corriente continua.

La figura 10 muestra la mejor solución obteni-

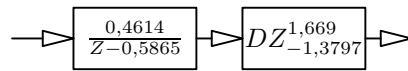
da (seleccionada en base al valor de error de validación de los experimentos realizados) mediante ambos esquemas. Se puede observar cómo ambos han sido capaces de identificar correctamente la estructura del modelo objetivo, diferenciándose únicamente en los valores identificados para los coeficientes numéricos. Se muestra también una comparación entre la respuesta real del motor (línea discontinua) y la respuesta de los mejores modelos hallados (línea continua) bajo cada enfoque. Ambos modelos reproducen fielmente la respuesta del motor.

VI. CONCLUSIONES

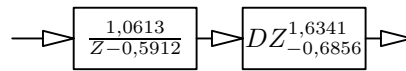
En esta trabajo se han comparado los esquemas de búsqueda estocástica SA-P, generalización del recocido simulado, y GA-P, un híbrido entre un algoritmo genético y un programa genético, al modelado de sistemas dinámicos con representaciones basadas en diagramas de bloques.

Los resultados son muy similares: en el caso del experimento realizado con el proceso sintético el GA-P supera ligeramente el rendimiento del SA-P (los errores tienden en general a ser más bajos y se estanca con menos frecuencia en mínimos locales), algo que ya no es tan claro (más bien podríamos hablar de que el SA-P supera al GA-P) en el caso del modelado del motor de corriente continua. Tales consideraciones se ven respaldadas por un análisis estadístico en base a diagramas de cuadros que permiten afirmar que no existen diferencias significativas entre ambos enfoques.

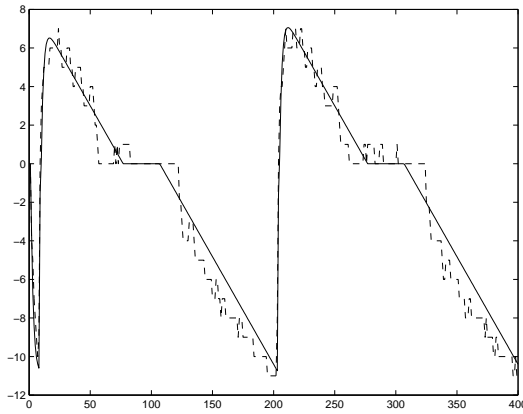
Esta similitud en términos de eficiencia no se corresponde con una similitud en otras posibles facetas de análisis. El algoritmo SA-P supera claramente al GA-P en términos de recursos computacionales, ya que usa un único individuo en el proceso al contrario del método evolutivo que se basa en poblaciones de varios cientos de individuos.



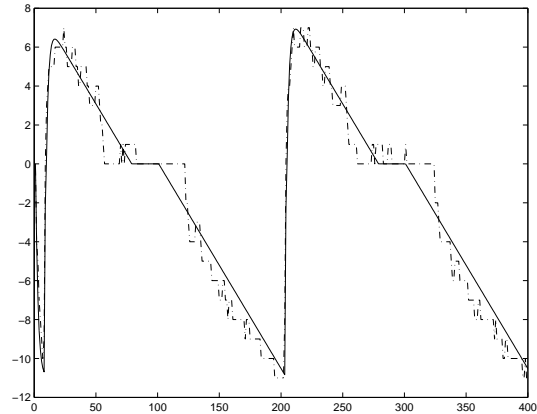
(a) Mejor modelo GA-P



(b) Mejor modelo SA-P



(c) Comparación entre las respuesta del motor y del mejor modelo hallado por el algoritmo GA-P



(d) Comparación entre las respuesta del motor y del mejor modelo hallado por el algoritmo SA-P

Fig. 10. Resultados del modelado del motor de continua

Asimismo, la estrategia de búsqueda del SA-P es también más sencilla, ya que se basa en un único operador de mutación mientras que en el GA-P se definen varios operadores genéticos, algunos de cruce y otros de mutación. Esta simplicidad hace también que el número de parámetros de diseño del algoritmo sea considerablemente inferior en el caso del SA-P, lo cual constituye una indudable ventaja al facilitar la experimentación, ya que el valor de los parámetros de diseño suele influir considerablemente en la eficiencia de la búsqueda y se suele ajustar habitualmente mediante un criterio de prueba-error.

En definitiva, considerando los puntos anteriores, se podría considerar preferible la aplicación del esquema SA-P, pues se trata de un algoritmo de mayor sencillez que aporta un rendimiento similar. No obstante, ha de tenerse en cuenta que se trata de una experimentación inicial, la cual será realizada de forma más exhaustiva en el futuro para poder establecer definitivamente su validez.

AGRADECIMIENTOS

Este trabajo ha sido desarrollado con el apoyo del Ministerio de Ciencia y Tecnología mediante el proyecto TIC2002-04036-C05-05.

REFERENCIAS

[1] H. Lopez A.M. Lopez and L. Sánchez. Ga-p based search of structures and parameters of dynamical

process models. In *7th Online World Conference on Soft Computing in Industrial Applications*, 2002.

- [2] H. Lopez A.M. Lopez and L. Sánchez. System inductive modelling using genetic programming with a genetic algorithm for parameter adjustment. In *15th IFAC World Congress on Automatic Control*, 2002.
- [3] V M Babovic, B Brozkova, M Mata, V Baca, and S Vanecek. System identification and modelling of vltava river system. In *Proceedings of the 2nd DHI Software User Conference*, 1991.
- [4] Saso Dzeroski, Ljupeo Todorovski, and Igor Petrovski. Dynamical system identification with machine learning. In Justinian P. Rosca, editor, *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications*, pages 50–63, Tahoe City, California, USA, 9 July 1995.
- [5] Anna I. Esparcia Alcazar and Ken C. Sharman. Some applications of genetic programming in digital signal processing. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1996 Conference Stanford University July 28-31, 1996*, pages 24–31, Stanford University, CA, USA, 28–31 July 1996. Stanford Bookstore.
- [6] Gary J. Gray, David J. Murray-Smith, Yun Li, and Ken C. Sharman. Nonlinear model structure identification using genetic programming. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1996 Conference Stanford University July 28-31, 1996*, pages 32–37, Stanford University, CA, USA, 28–31 July 1996. Stanford Bookstore.
- [7] Hugo Hiden, Mark Willis, Ben McKay, and Gary Montague. Non-linear and direction dependent dynamic modelling using genetic programming. In John R. Koza, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max Garzon, Hitoshi Iba, and Rick L. Riolo, editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 168–173, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.
- [8] Les M. Howard and Donna J. D'Angelo. The GA-P:

- A genetic algorithm and genetic programming hybrid. *IEEE Expert*, 10(3):11–15, June 1995.
- [9] Hitoshi Iba, Takio Karita, Hugo de Garis, and Taisuke Sato. System identification using structured genetic algorithms. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93*, pages 279–286, University of Illinois at Urbana-Champaign, 17–21 July 1993. Morgan Kaufmann.
 - [10] Martin A. Keane, John R. Koza, and James P. Rice. Finding an impulse response function using genetic programming. In *Proceedings of the 1993 American Control Conference*, volume III, pages 2345–2350, Evanston, IL, USA, 1993.
 - [11] Martin A. Keane, Jessen Yu, and John R. Koza. Automatic synthesis of both topology and tuning of a common parameterized controller for two families of plants using genetic programming. In Darrell Whitley, David Goldberg, Erick Cantu-Paz, Lee Spector, Ian Parmee, and Hans-Georg Beyer, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 496–504, Las Vegas, Nevada, USA, 10–12 July 2000. Morgan Kaufmann.
 - [12] A.M. Lopez, H. Lopez, and L. Sanchez. Graph based gp applied to dynamical systems modeling. *Connectionist Models of Neurons, Learning Processes and Artificial Intelligence*, 2084:725–732, 2001.
 - [13] Peter Marenbach. Using prior knowledge and obtaining process insight in data based modelling of bio-processes. *System Analysis Modelling Simulation*, 31:39–59, 1998.
 - [14] Peter Marenbach, Kurt D. Betterhausen, and Stephan Freyer. Signal path oriented approach for generation of dynamic process models. In John R. Koza, David E. Goldberg, David B. Fogel, and Rick L. Riolo, editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 327–332, Stanford University, CA, USA, 28–31 July 1996. MIT Press.
 - [15] B. McKay, B. Lennox, M. J. Willis, G. W. Barton, and G. A. Montague. Extruder modelling: A comparison of two paradigms. Technical report, Chemical Engineering, Newcastle University, UK, 1996. Appears in Control '96.
 - [16] R. Mead and J. A. Nelder. A simplex method for function minimization. *Computer Journal*, 7(4):308–313, 1965.
 - [17] Howard Oakley. Two scientific applications of genetic programming: Stack filters and non-linear equation fitting to chaotic data. In Kenneth E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 17, pages 369–389. MIT Press, 1994.
 - [18] L. Sánchez, I. Couso, and J. Corrales. Combining gp operators with sa search to evolve fuzzy rule based classifiers. *Information Sciences*, 1(3):175–191, 2001.
 - [19] A. H. Watson and I. C. Parmee. Identification of fluid systems using genetic programming. In *Proceedings of the Second Online Workshop on Evolutionary Computation (WEC2)*, number 2, pages 45–48, <http://www.bioele.nuee.nagoya-u.ac.jp/wec2/>, 4–22 March 1996. Nagoya University, Japan.
 - [20] Mark Willis, Hugo Hiden, Mark Hinchliffe, Ben McKay, and Geoffrey W. Barton. Systems modelling using genetic programming. *Computers in Chemical Engineering*, 21:S1161–1166, 1997. Supplemental.