15th Triennial World Congress, Barcelona, Spain

GA-P BASED MODELING OF NONLINEAR DYNAMIC SYSTEMS

Lopez, A.M. [*] **Lopez, H.** [*] **Sanchez, L.** [**]

[*] *University of Oviedo. Electrical Engineering Department. C. Universitario de Viesques, Ed. Departamental Nº 2, 33203, Gijon, Asturias, Spain.*
e-mail: [antonio,hilario]@isa.uniovi.es
[**] *University of Oviedo. Computer Science Department. C. Universitario de Viesques, Ed. Departamental Nº 1, 33202, Gijon, Asturias, Spain.*
e-mail: luciano@lsi.uniovi.es

Abstract:
Model construction is usually guided by a trial-error process, where each iteration is divided into two steps: (i) physical modeling and (ii) identification.
Genetic programming has been applied to automate this process in different ways. One of the most complete approaches is the described in project SMOG, where a set of model structures is evolved, being the set of parameters of each model optimized by means of classical methods. In this paper, a GA-P algorithm (a hybrid between genetic algorithms and genetic programming) is applied to the task permitting the evolution in parallel of model structures and parameters. Copyright ©2002 IFAC.

Keywords: Dynamic systems, nonlinear systems, identification algorithms, stochastic modelling, block diagrams, directed graphs, artificial intelligence, genetic algorithms.

1. INTRODUCTION

Genetic Programming has been applied to system modeling in different ways.

One of the initial solutions was that proposed in (Iba *et al.*, 1993), where a structured Genetic Algorithm, in a tree based representation, is used to solve identification problems. Quadratic functions of two input variables are used as the function set. Other interesting approaches are those described in (Babovic *et al.*, 1991) , (Sharman and Esparcia-Alcazar, 1993) and (Dzeroski *et al.*, 1995).

To our knowledge, the most complete application is described in project SMOG (Marenbach, 1998; Marenbach *et al.*, 1996), where the problem is addressed as a search of a model of a system using a block diagram representation, being the function set composed, among others, of continuous time blocks defined in the domain $\mathcal{S}$.

Nevertheless, this application presents some critical points. This paper analyzes them and proposes a solution, showing some results that surpass SMOG efficiency.

2. GENETIC ALGORITHMS, GENETIC PROGRAMMING AND THE GA-P ALGORITHM

Genetic Algorithms (Davis, 1991; Goldberg, 1989; Holland, 1975) are multi-point iterative stochastic optimization methods based on the natural evolution principle of *survival of the fittest*. Parameters in the target function are coded making up the individual. A set of randomly created individuals is evolved by means of the application of genetic operators until a

predefined termination criteria is satisfied. This operators are applied to individuals selected randomly from the current population with a bias in the direction of growing fitness.

Genetic Programming (Koza, 1992) is a derivation of the former. Individuals are, usually, tree based representations composed of terminals and functions, where the terminals (elements to provide information to the individual) and functions (elements to transform information into the individual) are specific to the field of application. Random creation and modification of individuals are adapted to the new representation. Fitness of each individual is evaluated using test data describing the desired behaviour of the solution structure.

Mixing both schemes, the **Genetic Algorithm Programmable**, or GA-P (Howard and D'Angelo, 1995), evolves in parallel a set of structures like the ones used in GP and a set of parameters of those structures. Genetic operators for the structural and parametric componentes are defined to fit the new scheme.

3. SMOG

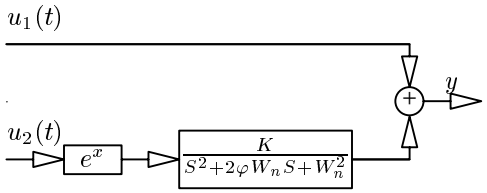
SMOG (Marenbach, 1998; Marenbach *et al.*, 1996) is an application of Genetic Programming to dynamical systems modeling. It evolves a set of diagram block representations in a tree representation (see figure 1), where the terminal set is composed of the input variables and the function set is composed, among others, of linear S-blocks, non-linear blocks, and the usual arithmetic operators found in block diagram representations such as $+$ and $-$.

Any individual in the population will undergo a parameter adjustment by means of Hooke-Jeeves algorithm (Hooke and Jeeves, 1961).

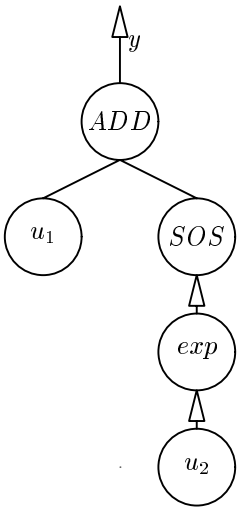
Fitness calculation of models (based on sampled data) is made after its conversion to the discrete time domain.

SMOG is an interesting application of GP to system modeling. It allows to analyze the solution, a great advantage over other learning methods sometimes rejected in the industrial environment due to their black box nature. Nevertheless, SMOG presents some drawbacks:

- It uses a tree based representation. This makes it impossible to model a wide set of systems, such as those involving nested or non-unitary feedbacks. The reason is that a block diagram is not a tree when it includes feedback, but a directed graph. So, a graph based representation for models is used.
- Objective function to minimize (residual error between the model and real system output) presents local minimum (A.M. Lopez and Sanchez, 2001), making classical optimization methods applied to the identification of models involved in the process invalid. As shown in previous studies (A.M. Lopez and Ojea, 2000), genetic algorithms provide better results concerning the parsimony of generated models and preventing the tool from being easily trapped by local minima. Nevertheless, GAs could be highly consuming in terms of computational resources if applied to each individual in the population. Our application tries to evolve in parallel the structured model by means of a GP and the vector of parameters by means of a GA, making up a GA-P based modeling tool.



(a) A sample system



(b) Tree based representation

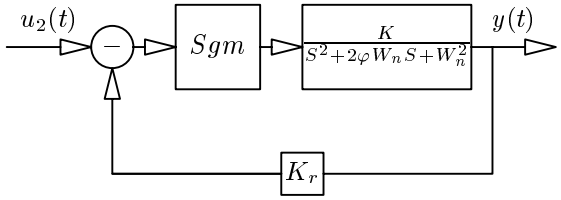
Fig. 1. Tree based representation of a block diagram

4. TOOL DEFINITION

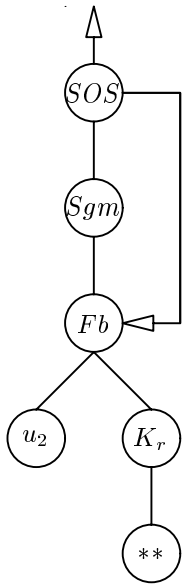
Base class system definition Nonlinear dynamic systems are defined as such base class. It would make not much sense to apply a high consuming tool like GP to linear dynamic modeling, where several methods of high efficiency have been developed over the years.

Individual Representation Each model is represented by a two component structure. One is a graph based representation of the model structure, and the other is a vector of real values containing its parameters.

- Structural component. The graph based representation proposed (see figure 2), mixes a



(a) A sample System



(b) Graph based representation

Fig. 2. Graph based representation of a block diagram

link nodes list with ideas from (Marenbach *et al.*, 1996) and (Sharman and Esparcia-Alcazar, 1993).

A special feedback node, different from the one defined in SMOG, is used, hanging from it the input and the feedback branches. It also contains a third link to another node in the graph from which the feedback signal will be taken. This pointed node will play, together with its own function, the role of the bifurcation node.

Algebraic loops are forbidden. So, a unit delay is implicitly included in the feedback branch. Also, physical systems never respond instantly to an excitation. So, a unit delay is implicitly linked to the output of the graph and dynamic blocks used respond instantly.

- Parameters component. It is composed of a vector of real values containing the parameters of the model to be evolved by the GA component of the algorithm. It was decided to use a real value codification based on (Davis, 1991), where it is said that the best codification you can use is the most natural for the problem, adapting the GA to use this codification.

Individual Evaluation Fitness calculation is based on the residual error between the system and model outputs. Three samples of the system are used for the evaluation of the individual, making up three different fitness measures. The first (training) will be used to evolve the structural component, the second (test) to evolve the parameters component and the third (validation) will be used as an unbiased measure of the error in order to determine its generalization capabilities.

Genetic Operators Besides the reproduction operator, two sets of genetic operators are defined. One affects the structural component of individuals and the other affects their parameter component. A generational approach is used. So, after the application of genetic operators a new full population is defined to replace the actual one.

- Structural Genetic Operators. Subtree crossover (Koza, 1992) and internal crossover (Kinnear, Jr., 1994) are used. Subtree, node and feedback mutation operators are also used. This set of operators only affect the structural component of the individuals involved, not the parametric one. Any application of a structural genetic operator could give as a result an invalid individual. In that case, a reparation strategy is applied to it.
- Parameter Genetic Operators. Two structurally identical individuals are selected from the population for each application of this set of operators. They only affect their parameter component, not the structural one. Real based genome crossover operator is defined for the parameters of the model as a random movement of a vector in the direction of the other. After crossover, a mutation, a direct search or both can be applied to the resulting offsprings depending on predefined probabilities. Mutation is defined as a crossover with a randomly generated individual. Direct search is performed by means of Nelder & Mead algorithm (Mead and Nelder, 1965) run for a few iterations.

5. AN EXAMPLE OF APPLICATION

A modeling of a synthetic system shown in figure 3 is done.

5.1 *Algorithm Preparation*

As the function set used for the modeling a linear dynamic first order system, an arithmetical operator of addition, a feedback node, a static gain and a saturation block will be used. The tool will try to combine them to form an “optimal” model of the

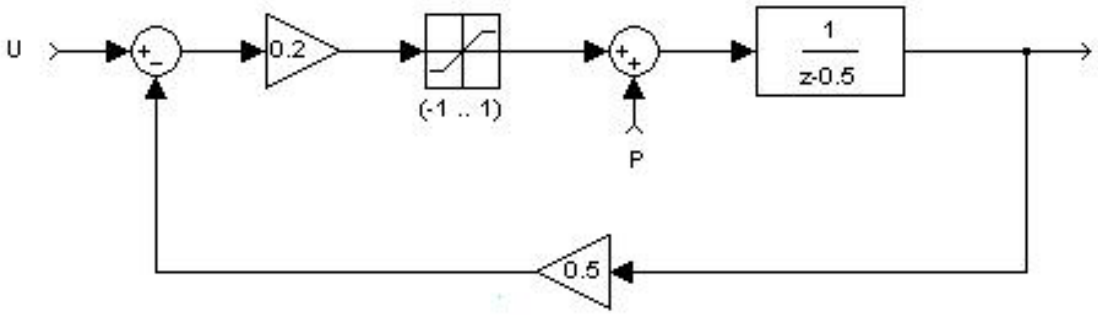


Fig. 3. System to model

Table 1. Variation ranges for parameters of functions used in the example of application

	Minimum	Maximum
First order static gain	1.5	2.5
First order pole	0.25	0.75
Static gain	0.1	0.75
Saturation	-1.5	1.5

Table 2. GA-P parametrization for the example of application

Parameter	Value
Generations	50
Individuals	501
Reproduction Fraction	0.1
Parametric Crossover fraction	0.2
Parametric mutation probability	0.05
Direct Search probability	0.5
Iterations of Direct Search	10
Structural Crossover Fraction	0.65
Internal Crossover Fraction	0.05
Structural Mutation Fraction	0.02
Node Mutation Fraction	0.02
Feedback Mutation Fraction	0.01

system, departing from a randomly set of models by the application of the genetic operators defined.

Also, it is necessary to specify a range of variation for each parameter of the different components of the function set. Table 1 shows those used at this experiment. Parametrization of the algorithm is shown in table 2.

5.2 *Results*

Experiment is repeated 10 times. Table 3 contains the numerical results (training, test and validation fitness measures for each experiment, together with their average over the 10 experiments). Errors are low, so models are quite good.

Regarding the analytical solutions, best model was obtained at experiment N^o 10 (see figure 4). Structure of the model is the same as that of the real system. Only a deviation is present in the parameters vector, a problem which could be easily solved applying a more intensive identification algorithm over this structure.

Table 3. Modeling errors. GA-P algorithm.

Experiment	Training	Test	Validation
1	0.0060	0.0049	0.0038
2	0.0078	0.0084	0.0073
3	0.0065	0.0049	0.0041
4	0.0062	0.0050	0.0035
5	0.0125	0.0118	0.0137
6	0.0062	0.0056	0.0042
7	0.0202	0.0240	0.0102
8	0.0060	0.0052	0.0043
9	0.0381	0.0515	0.0342
10	0.0015	0.0013	0.0015
Average	0.0111	0.0123	0.0087

5.3 *Comparison with SMOG*

SMOG is applied to model the same system under similar conditions. In order to make a fair comparison, the same number of evaluations of the objective function is allowed. Although it could seem that the time involved in the modeling (directly proportional to the number of evaluations of the objective function) is not very important, it is of great importance because any application of an evolutive method is usually repeated several times.

The GA-P application made 48096 evaluations of the objective function. SMOG is going to be applied allowing to perform 50000 evaluations of it. A population size of 51 individuals will be used. The algorithm runs for 20 generations. Hooke-Jeeves method will be applied to identify each system for 54 iterations. A reproduction rate of 0.1, a crossover rate of 0.8 and a mutation rate of 0.1 will be used. Experiment is also repeated 10 times.

Table 4 contains the results (training, test and validation errors for each experiment, together with their average over the 10 experiments). Errors are low, but they are higher than those obtained by means of the application of the GA-P algorithm.

Best model was obtained at experiment number 3 (see figure 5). Structure is very different from that of real system, providing little information about it. From this experiment, it can be concluded that the tool is easily trapped by local minima.

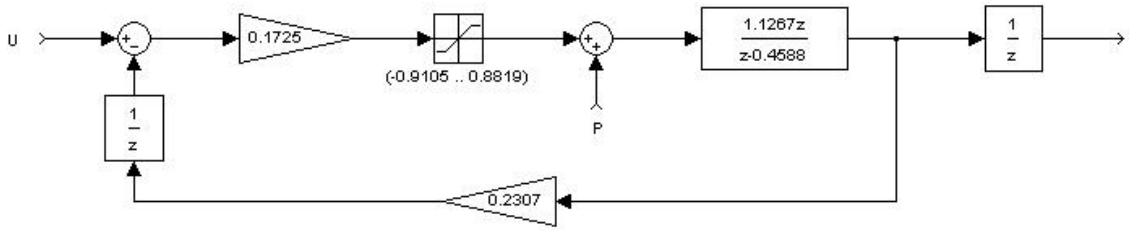


Fig. 4. GA-P best result

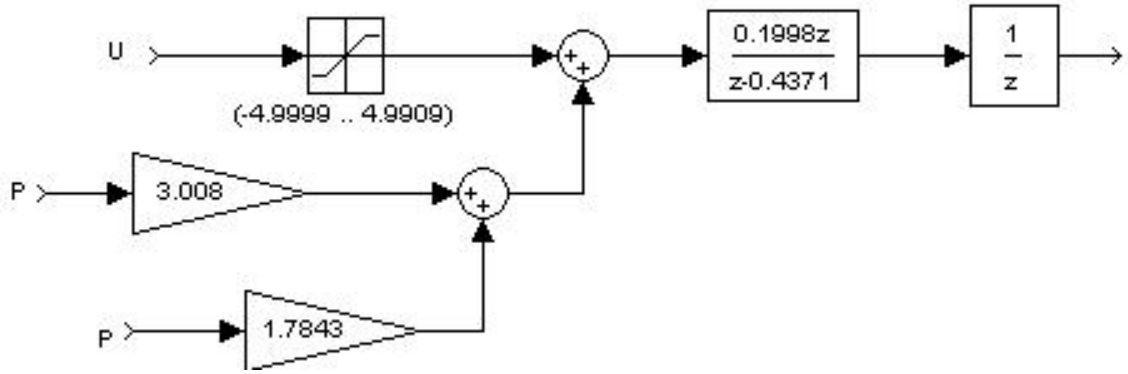


Fig. 5. SMOG best result

Table 4. Modeling errors. SMOG algorithm

Experiment	Training	Test	Validation
1	0.0029	0.0029	0.0015
2	0.0721	0.0695	0.0638
3	0.0025	0.0026	0.0015
4	0.0059	0.0057	0.0053
5	0.0038	0.0030	0.0028
6	0.0040	0.0040	0.0027
7	0.0364	0.0343	0.0372
8	0.0064	0.0045	0.0042
9	0.0049	0.0034	0.0031
10	0.0071	0.0064	0.0052
Average	0.0146	0.0136	0.0127

6. MODELING A REAL PROCESS

As a final validation, a real process was modeled under the proposed GA-P approach. A simple target system was selected in order to be able to contrast the GA-P result with a known model of the system, thus analyzing the capacity of the algorithm to propose models with significance. Such system was a direct current motor.

An open loop direct current motor is usually modeled as a first order dynamic system. Another pole would be present in a more precise model, but its effect is normally ignored. Also, at low voltage values, the motor usually presents a dead zone.

Three different samples for modeling were taken from the motor by means of squared, triangular and random input signals. In order to make the dead zone to have an appreciable effect in the behaviour of the system, the motor was fed at low voltage values.

The function set used in this experiment was composed of first and second order dynamic subsystems, a static gain and a dead zone block. Variation ranges for parameters of subsystems are shown in table 5. Parametrization of the algorithm is shown in table 6.

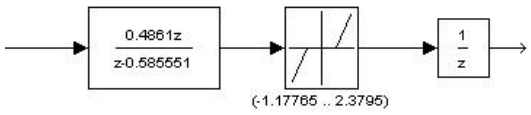
Table 5. Variation ranges for parameters of functions used in the real process application

	Minimum	Maximum
First order static gain	0.5	1.5
First order pole	0.25	0.75
Second order static gain	0.5	1.5
Second order pole (real component)	0.25	0.75
Second order pole (imaginary component)	0	0.25
Static gain	0.5	1.5
Dead Zone	-10	10

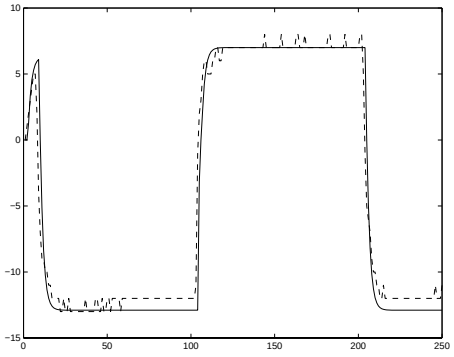
Table 6. GA-P parametrization for the real process application

Parameter	Value
Generations	100
Individuals	1001
Reproduction Fraction	0.1
Parametric Crossover fraction	0.6
Parametric mutation probability	0.05
Direct Search probability	0.5
Iterations of Direct Search	10
Structural Crossover Fraction	0.2
Internal Crossover Fraction	0.05
Structural Mutation Fraction	0.03
Node Mutation Fraction	0.02
Feedback Mutation Fraction	0

As a result of the experiment, a model composed of a cascade association of a first order dynamic system and a dead zone of a few r.p.m. was found (see figure 6(a)). In figure 6(b) a comparison between the motor (plotted line) and model (continuous line) responses is shown. System response is accurately reproduced.



(a) Motor model



(b) Comparison of motor (plotted line) and model (continuous line) responses.

Fig. 6. Modeling a direct current motor

7. CONCLUSIONS

A new scheme is proposed to model physical systems based on the Genetic Algorithm Programmable. This new scheme proposes a graph based representation for the structural component and a real value vector for the parameters component. Genetic operators are defined to fit both components representation.

The scheme has been applied to model a synthetic nonlinear dynamic system, showing the capacity to provide concise and precise solutions. Also, the structure of the system was identified, allowing to conclude a high efficiency to defect local minima.

A comparison with SMOG under similar conditions has been made. Modeling errors are higher and the solution is structurally far from the real system, being this approach easily trapped by local minima.

Also, the proposed approach has been applied to model a real system. Result is coherent with a known model for the system and it accurately reproduces the behaviour of the real system.

8. REFERENCES

A.M. Lopez, H. Lopez and G. Ojea (2000). Dealing with parameter tuning precision under a gp based dynamical system modeling tool with a ga for parameter adjustment. In: *Proceedings of INDUSCON 2000*.

A.M. Lopez, H. Lopez and L. Sanchez (2001). Graph based gp applied to dynamical systems modeling. *Connectionist Models of Neurons, Learning Processes and Artificial Intelligence* pp. 725–732.

Babovic, V M, B Brozkova, M Mata, V Baca and S Vanecek (1991). System identification and modelling of vltava river system. In: *Proceedings of the 2nd DHI Software User Conference*.

Davis, Lawrence (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold. New York, NY.

Dzeroski, Saso, Ljupeo Todorovski and Igor Petrovski (1995). Dynamical system identification with machine learning. In: *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications* (Justinian P. Rosca, Ed.). Tahoe City, California, USA. pp. 50–63.

Goldberg, David E. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison-Wesley. Reading, MA.

Holland, John H. (1975). *Adpatation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbor, MI.

Hooke, R. and T. Jeeves (1961). Direct search solution of numerical and statistical problems. *Journal of the Association for Computing Machinery* (8), 212–229.

Howard, Les M. and Donna J. D'Angelo (1995). The GA–P: A genetic algorithm and genetic programming hybrid. *IEEE Expert* **10**(3), 11–15.

Iba, Hitoshi, Takio Karita, Hugo de Garis and Taisuke Sato (1993). System identification using structured genetic algorithms. In: *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93* (Stephanie Forrest, Ed.). Morgan Kaufmann. University of Illinois at Urbana-Champaign. pp. 279–286.

Kinnear, Jr., Kenneth E. (1994). Alternatives in automatic function definition: A comparison of performance. In: *Advances in Genetic Programming* (Kenneth E. Kinnear, Jr., Ed.). Chap. 6, pp. 119–141. MIT Press.

Koza, John R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press. Cambridge, MA, USA.

Marenbach, Peter (1998). Using prior knowledge and obtaining process insight in data based modelling of bioprocesses. *System Analysis Modelling Simulation* **31**, 39–59.

Marenbach, Peter, Kurt D. Betterhausen and Stephan Freyer (1996). Signal path oriented approach for generation of dynamic process models. In: *Genetic Programming 1996: Proceedings of the First Annual Conference* (John R. Koza, David E. Goldberg, David B. Fogel and Rick L. Riolo, Eds.). MIT Press. Stanford University, CA, USA. pp. 327–332.

Mead, R. and J. A. Nelder (1965). A simplex method for function minimization. *Computer Journal* **7**(4), 308–313.

Sharman, Ken C. and Anna I. Esparcia-Alcazar (1993). Genetic evolution of symbolic signal models. In: *Proceedings of the Second International Conference on Natural Algorithms in Signal Processing, NASP'93*. Essex University, UK.