Genetic Search of Block-Based Structures of Dynamical Process Models

A. López[1], L. Sánchez[2]

Depts. of Ingeniería Eléctrica (1) and Informática (2). C. Universitario de Viesques.
Ed. Departamentales, 33204, Gijón, Asturias.
e-mails: `antonio@isa.uniovi.es`, `luciano@ccia.uniovi.es`

Abstract. Genetic identification of models of dynamical systems is becoming a well stablished research field. Nowadays it is hard to obtain more precise numerical results than *state of the art* methods, but, in our oppinion, there is still room to improve the understandability of genetically induced models. In this paper it is proposed a method that focuses in the comprehensibility of the final model, while keeping most of the numerical precision of former studies.
The main innovation in this work is centered in the concept of "understandable" system. We do not use state space designed, rule based models, but z-transform based models, comprising linear, discrete dynamical models of first or second order and memoriless nonlinear elements (saturation, dead zone or other nonlinear gains.) This way, we provide control engineers with their prefered representation in moderate to complex models, and facilitate the task of designing control systems for these processes.
Keywords: Block-based models, Process control, GA-P algorithms, System Identification.

1 Introduction

Genetic identification of models of dynamical systems is becoming a mature research field. It is widely admitted than both GA and GP can find very precise representations of dynamical models. But the use of the structures found with GP algorithms is not very extended in practice. This is mostly due to the complexity of the final output: the output of canonical GP, even in simple nonlinear models [7] is hard to interpret, and not very different in complexity from a neural network.

The comprehensibility of a model of a dynamical system has also been studied under different perspectives. In our oppinion, the most advanced methods have been developed in the fuzzy community (see, for instance [2]). Genetic Fuzzy systems are able to produce a set of linguistically understandable fuzzy rules that define the state-space behavior of a dynamical system. Fuzzy models have important advantages over black-box models like the formerly mentioned neural networks, or least-square fitted polynomials. But there are many circumstances in which control engineers are not confortable with a state space, rule based description of a dynamical model. According to our own experience, many control systems are best designed in the classical framework (z transform). The prefered output for a small to moderately complex dynamical model is a block-based structure where blocks are either

- A linear, discrete dynamical model of first or second order.
- A memoriless nonlinear element (saturation, dead zone or other nonlinear gain).

In fact, there are preliminary works in GP in which is used an structure with certain similarities to the one being proposed herein [8,9]. But the work in this field, up to our knowledge, was not continued. In this paper we propose to use more modern hybrid genetic techniques than those used in these preliminar works and to improve both numerical properties and the quality of the block based representation. Our objective is to produce models with a high linguistic quality, and to promote their practical use.

1.1 Structure of the Paper

The outline of this paper is as follows: in Sect. 2, different genetic-based models of dynamical systems are discussed. In Sect. 3 the algorithm proposed here is described. Then (Sect. 4) an experimental validation of our proposal is done, modeling both a synthetic and a real process and comparing the results with those obtained with previous works. The paper finishes (Sect. 5) with the concluding remarks and future work.

2 Genetic-based comprehensible models

Most of the evolutionary methods for system identification from sampled data focus in nonlinear state space-based models. For this kind of models, the objective of the learning process is the production of a set of difference equations defining the dynamics of the process. Unfortunately, for practical purposes, a set of equations that relates all state variables between them is hard to manage in all but small sized problems. Modular representations are usually preferred, because they allow to determine groups of variables affected by specific parameters.

Genetic Programming has been applied to learn such modular models. One of the first examples was given in [6], where a structured Genetic Algorithm, in a tree based representation, is used. The set of functions that was proposed contained only two-input quadratic functions, which are not the building blocks that control engineers expect to find in structured models. Some implementations nearer to usual practice can be found in [3,7] and other, less common approaches to model the dynamics of a system, are described in [4]. Most of these schemes introduce dynamic considerations by means of extended terminal sets, that include either input and input-output delayed variables.

One of the most complete methods is described in SMOG [8,9]. The problem is addressed there as a search of a diagram block based representation of a

```
1. Initialize random population of models.
2. Tune parameters of models (Hooke-Jeeves algorithm).
3. Calculate fitness.
4. Selection of models and application of genetic operators.
5. Go to 2).
```

Fig. 1. SMOG evolution. Canonical GP is used for structural search and Hooke-Jeeves method is used for parameter tuning

model of the process in a tree codification. Under the considered approach (see Fig. 1), hierarchical evolutionary algorithms are applied: canonical GP is used for the evolution of model structures and combined with deterministic numerical optimization methods (Hooke and Jeeves algorithm) for parameter tuning. An iterative search of structure and parameters is done: each model considered is parametrically tuned by means of Hooke-Jeeves algorithm as a previous step to fitness evaluation. Genetic operators defined for evolution affects only the structure of the models.

We will show in this paper that, according to our experimentation, better results can be obtained if a new representation in which the search is not done hierarquically, but by means of an hybrid method between GA and GP (the GA-P algorithm, [5]). GA-P is able to search *in parallel* in both structure and parameter spaces. It is remarked that we use the word "hierarquical" to describe an algorithm in which GP operators are used first, to find an structure, and then a local search, classical or genetic, is launched to optimize the parameters of that structure. In other words, the GP serves to find structures, but the numbers found in the terminal nodes only serve as a starting point for the parameter search.

3 Proposed Algorithm

Our algorithm will evolve a set of diagram block representations of the process. A diagram block is, in turn, a series, parallel or feedback association of subsystems. Series association is intrinsic, and parallel association will be allowed by means of arithmetic operators, such as + and -, and feedback representation will be allowed by means of an special operator [1] described next.

Regarding the catalog of subsystems, we used only memoryless version of the common non-linear features of physical systems, such as dead zones or saturations. All the dynamic behavior is delegated to linear elements: we include in the function set a reduced group of linear models (first and second order dumped linear subsystems, unitary delay and static gain) such that it is possible to get higher order systems by means of series association.

3.1 GA-P Algorithms

GA-P [5] is an hybrid between genetic algorithms and genetic programming, that was first used in symbolic regression problems. Individuals in GA-P have two parts: a tree based representation and a chain of numerical parameters. Different from canonical GP, the terminal nodes of the tree never store numbers but linguistic identifiers that are pointers to the chain of numbers.We do not use the initial implementation of GA-P but rather a niche-based strategy described in [10].

3.2 Representation

The parameter part of the GA-P is an standard, real coded vector of values. The structure has to be adapted to the problem at hand. Otherwise, the tree based representation of GA-P would make it impossible to model a wide set of systems, such as those involving nested or not unitary feedbacks. The reason is that a block diagram is not a tree when it includes feedback, but a directed graph. The proposed representation circumvents this problem by means of a special feedback node. Both input and the feedback branches originate in it. The terminal nodes of the feedback branch (marked as "**") are recessive. This way, standard structural modification operators can be applied at any point in the individual to evolve structures.

It also contains a third link to another node from which the feedback signal will be taken, converting the representation in a graph. This pointed node must be contained in the path between the feedback node and the output node of the system. Otherwise, feedback node looses its significance. This consideration must be present in the creation and modification of individuals as a consequence of structural genetic operators. When an individual does not accomplish this condition after an structural modification, invalid feedback nodes are reinitialized.

Algebraic loops are neglected by means of the implicit inclusion of a unit delay in the feedback branch. To prevent series associations of delays, dynamic blocks used respond instantly. But, known the fact that physical systems never respond instantly to an excitation, a unit delay is also implicitly linked to the output of the model.

3.3 Genetic Operators

Two sets of operators are applied in the evolutionary process: structural and parameter genetic operators. Again, in the parameter part a fairly standard aithmetic crossover, followed by a local search adjust. The structural operators (GP crossover) must take care of the special tree-based encoding of a graph that is made here, thus feedback links are inhibited during the process.

4 Numerical Results

To validate our approach, as a first test, an empirical control system of a first order process with a proportional saturated controller and a sensor without dynamics (see Fig. 3(a)) was modelled by means of the defined GA-P strategy.

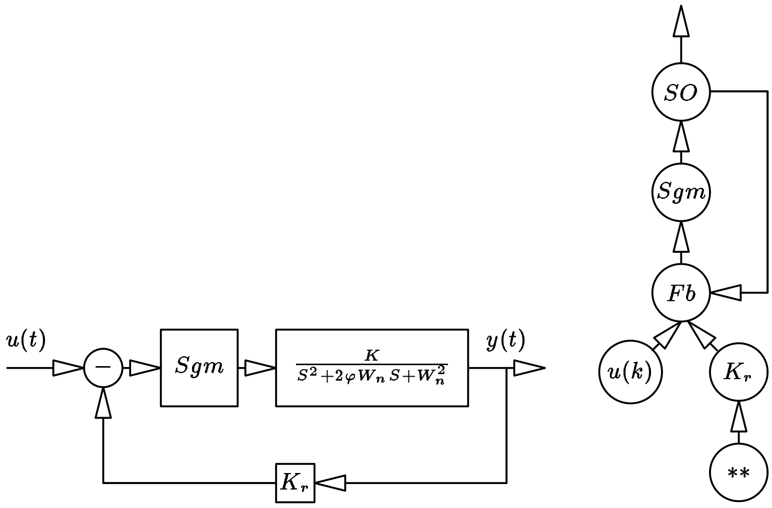


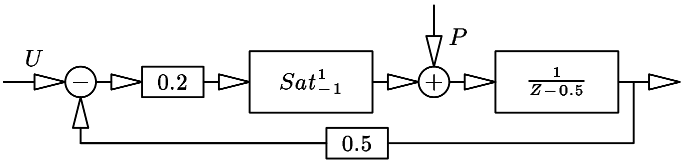
Fig. 2. Block diagram representation of a feedback system (left) and its genetic representation (right.) "SO" stands for "Second Order" and "*Sgm*" for "sigmoid function". Also, "**" stands for a recessive terminal

It was also compared with a hierarchical process. Both approaches were stopped after certain number of evaluations of the objective function.

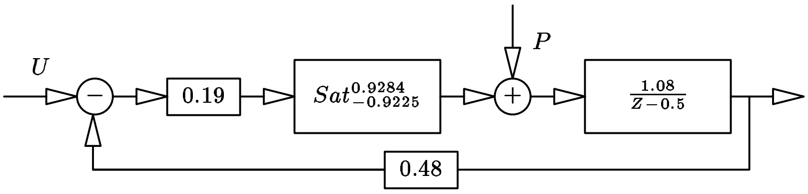
Experiments were repeated 10 times each. Table 1 contains validation errors for each experiment. Observe that GA-P improves slightly the results, but the differences are not significant. The gain with GA-P is in the identified structure (see Figs. 3(b) and 3(c), where the best models obtained by both approaches are shown). In this case, GA-P found exactly the structure of the target model, explaining very well the data relationships. In contrast, the hierarchical method was trapped in a local minimum of the structure. It is only capable of fitting the sampled data.

As a final test, a real system was modelled by means of the proposed scheme. A DC motor was selected, in order to have information enough to contrast the GA-P solution with a known model for the process (usually a first or second order dumped linear subsystem with a non-linear dead zone component).

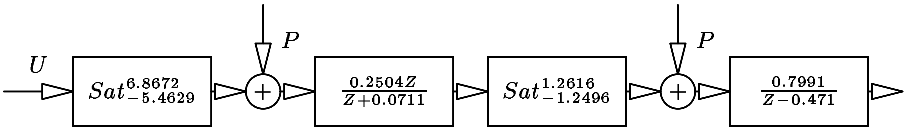
Experimental conditions were the same as in the preceding section. Table 1 contains the numerical validation errors for each experiment. From it, it can be concluded that the best result was found at experiment 10, shown in Fig. 4. Solution is close to a known model for the system: the search scheme is capable of capturing the most significant relationships in the data. This figure also includes a comparison between the motor and the model responses using a squared input signal. Observe that the behavior is correctly reproduced and the noise is smoothed as expected.



(a) Target Model



(b) GA-P Search Best Model



(c) Hierarchical Search Best Model

Fig. 3. Modelling a synthetic example. Upper part: target model. Central and lower parts: structures of the learned models. "Sat_a^b" stands for "saturation" block with limits in a, b

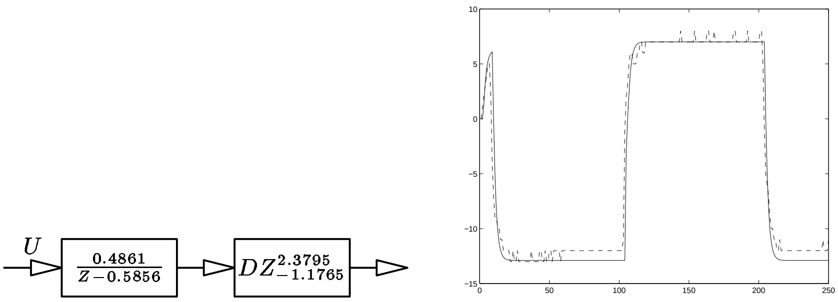


Fig. 4. Modelling of a direct current motor. Left: best model found ("DZ_a^b" stands for "dead zone" block with limits in a, b). Right: Comparison of model (continuous line) and system (plotted line) responses

Table 1. Numerical results: GA-P (left) and hierarchical (center) modelling errors for the synthetic problem. Right table: modelling errors for the direct current motor

Experiment	Error
1	0.00017
2	0.0004
3	0.00005
4	0.00004
5	0.00019
6	0.00005
7	0.00005
8	0.00006
9	0.00029
10	0.00007
Average	0.00014

Experiment	Error
1	0.00206
2	0.00301
3	0.00129
4	0.00184
5	0.00287
6	0.00112
7	0.00111
8	0.00107
9	0.00147
10	0.00263
Average	0.00185

Experiment	Error
1	0.9196
2	0.7755
3	0.7354
4	0.8433
5	0.9223
6	0.9259
7	1.1809
8	1.0134
9	1.0976
10	**0.6933**

5 Concluding Remarks

We have shown that the use of hybrid methods can improve the understandability of the model of a dynamical system. A parallel evolutive search of parameters and structure was proposed, and it was shown that it did not waste time optimizing parameters for invalid structures neither discards structures too early, thus the quality of the output was improved.

Perhaps the main interest in this work is in the concept of "understandable" system. We did not use state space designed, rule based models, neither we used ordinary GP arithmetical expressions, but z-transform based models, comprising linear, discrete dynamical models of first or second order and memoriless nonlinear elements (saturation, dead zone or other nonlinear gains.) This way, we provide control engineers with their prefered representation in moderate to complex models.

Acknowledgements

This work was funded by Spanish Ministery of Science and Technology and by FEDER funds, under the grant TIC-04036-C05-05.

References

1. H. Lopez A.M. Lopez and L. Sanchez. Graph based GP applied to dynamical systems modeling. *IWANN 2001. Connectionist Models of Neurons, Learning Processes and Artificial Intelligence*, pages 725–732, 2001.
2. Cordón, O., Herrera, F. (2000) "A proposal for improving the accuracy of linguistic modeling". IEEE Transactions on Fuzzy Systems, 8, 3, pp. 335-344.

3. G.J. Gray, D.J. Murray-Smith, Y. Li, and K.C. Sharman. Nonlinear model structure identification using genetic programming. In *Late Breaking Papers at the Genetic Programming 1996 Conference*, pages 32–37, Stanford University, CA, USA, 1996.
4. H. Hiden, M. Willis, B. McKay, and G. Montague. newblock Non-linear and direction dependent dynamic modelling using genetic programming. In *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 168–173, Stanford University, CA, USA, 1997.
5. L.M. Howard and D.J. D'Angelo. The GA–P: A genetic algorithm and genetic programming hybrid. *IEEE Expert*, 10(3):11–15, June 1995.
6. H. Iba, T. Karita, H. Garis, and T. Sato. System identification using structured genetic algorithms. In *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93*, pages 279–286, University of Illinois at Urbana-Champaign, 1993.
7. M.A. Keane, J.R. Koza, and J.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.
8. P. Marenbach. Using prior knowledge and obtaining process insight in data based modelling of bioprocesses. *System Analysis Modelling Simulation*, 31:39–59, 1998.
9. P. Marenbach, K.D.. Betterhausen, and S. Freyer. Signal path oriented approach for generation of dynamic process models. In *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 327–332, Stanford University, CA, USA, 1996.
10. L.A. Sanchez and J.A. Corrales. Niching scheme for steady state GA-P and its application to fuzzy rule based classifiers induction. *Mathware and Soft Computing*, 7(2-3):337–350, 2000.