EVOLUTIONARY METHODS FOR DESIGN, OPTIMISATION AND CONTROL
K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou and T. Fogarty (Eds.)

CAUSALITY AND DESIGN OF DYNAMICAL SYSTEMS

I. Santibáñez Koref and I. Boblan⋆
Technische Universität Berlin
Bionik und Evolutionstechnik
Ackerstr. 71–76, 13355 Berlin, Germany
e-mail: isk@bionik.tu-berlin.de

F. Lohnert and A. Schütte
DaimlerChrysler
FuE, Kognition und Robotik
Alt Moabit 96a, 10559 Berlin , Germany
e-mail: frieder.lohnert@daimlerchrysler.com

Abstract. We present a software for the generation of dynamical systems. The main idea of this software is to introduce the concept of strong causality in generation procedure. Using a simple example we show how the different parts of the software work together.

Key words: Evolution Strategy, Genetic Programming, Control, System identification

1 INTRODUCTION

The design of dynamical systems for industrial applications can be divided in to two parts: first the identification of a plant or system and second the design of an appropriate controller. This paper presents a software tool that unifies identification and design task. Since identification and design are different tasks, they can be reduced to common search problem: Find a dynamical system that fulfills certain required properties.

This search problem, can be solved using an optimization algorithm. In this paper we show how to combine and adapt two evolutionary optimization algorithms to find optimal solutions. As evolutionary optimization algorithms we use Genetic Programming and Evolution Strategy.

We implemented this approach in the SCADS (Strong CAusality driven generation of Dynamical Systems) system. Another the main property of SCADS is that one basic concept behind the implementation is the use of "Strong Causality"[7] through all stages of the search process. "Strong Causality" means that small variations of the individual, i.e. dynamical system, should produce small variations of the quality.

Genetic Programming (GP)[4] is used for the structure identification of the dynamical system. The GP is adapted by implementing strong causality in the representation and in the genetic operators.[6] The Evolution Strategy (ES).[7] is used to adapt the parameters of the dynamical systems. Strong causality in this algorithm means that the quality assignment to each system should be strong causal.

One of the main aspects of SCADS is the practical usability. That's why we build it up using Mathworks inc. MATLAB/Simulink. There are a lot of simulation systems which are in practical use in a wide range of fields. One of these systems is MATLAB, which offers wide variety of tools for the design and identification of dynamical systems. Specially a number of toolboxes offers well proven applications and tailored methods for the simulation of dynamical systems.

In the next section we give an outline how to optimize dynamical systems using evolutionary algorithms. Then we present the different possibilities to introduce strong causality.At the end we present a small example in order to see how SCADS can be used and draw some conclusions from the presented work.

2 Evolution of dynamical systems

Our approach for the search of dynamical systems is to embed the search in an evolutionary process. Evolutionary algorithms (EA) have been used for a number of applications related to dynamical systems.[4,7] An evolutionary algorithm can be described as a so called evolution loop.

First we have to formulate an appropriate description of dynamical systems. That means that the description should be evolvable and be able to describe a wide variety of dynamical systems. We choose a presentation as a directed graph.[6] This special representation as a graph guarantees that it is possible to create closed loops in the dynamical systems, thus it is possible to generate the most common structures for dynamical systems. The nodes represent the standard building blocks for dynamical systems (e.g. P-,PD- ,PID-blocks etc.[2]). The vertices represents the signal flow between this blocks. One individual is a dynamical system, represented as a graph. That means, that we are evolving graphs.

As an important feature it is necessary to assign to each individual a quality. In our case quality means: How well does the dynamical system fulfill the required task. In case of system identification, how well does it reproduce a given behavior. In the case of controller design: how well does the system control a given plant. In order to assign a suited quality the user has to provide a calculation procedure, which involves the simulation of the dynamical system by a integration algorithm. As we use standard representation for functional blocks, we have to take in to account that the blocks are parametrized. Thus not only the structure of the dynamical system is important, also the parameters of the blocks are significant for the resultant system behavior. We have to choose the appropriate parameter settings in order to able to compare two systems. Thus it is necessary to optimize the parameters in order that the quality criterion is fulfilled as best as possible. That means that besides a simple simulation of the dynamical system is it necessary to optimize the parameters of this system in order to determine the quality of the system.

Providing the quality information it is easy to select the best individuals. The first choice would be to use truncation selection or tournament selection. But taking in to account that for each quality assignation a simulations are needed, we decide to choose another selection structure. For selection we take a certain number of individuals from the population, select the best from this group and replace the worst individuals by new generated individuals (called partial tournament selection). Thus we can guarantee that we have not a generational EA. That guarantees a better usage of available parallel resources. Using this approach for selection we can expect an efficient implementation of the EA. That is why we implement the quality assignment on independent, asynchronous computational slaves, which guarantee a linear speed up of the execution time.

3 STRONG CAUSALITY

The next problem is how to define a strong causal quality function. This quality is decisive for the efficiency of the whole algorithm. Figure 1 presents two functions for quality assignment for control systems that depend on one parameter. The continuous line represent a quality function that tends to have big value changes with small parameter changes. The function is multimodal and hard to optimize for a wide variety of optimization algorithms. The second function depicted in figure 1 is a so called strong causal function (dotted line). We can see that the function is smoother, thus small changes in the parameter gives also small changes in the quality. The second property of this function is that it is unimodal, thus it can be optimized very easy. The general notion of strong causality[1,7] is that small changes in the parameters should lead to a small change in the quality of the object which is evaluated.

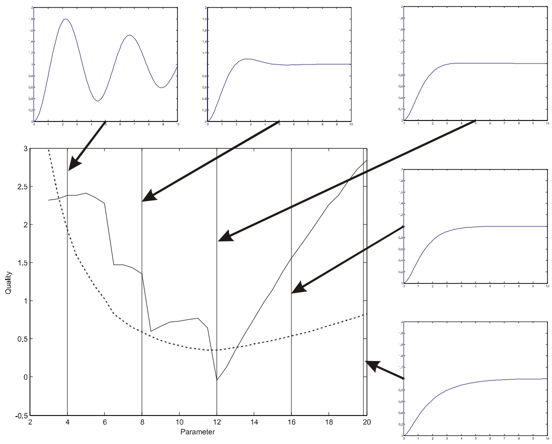

Figure 1: Comparison between a Strong Causal (doted line) and a non Strong Causal quality function for simple control system.

In SCADS the user has to supply a quality function. If this function is strong causal or not has to be scrutinized by the user. Some examples of how to design strong causal function can be found in some papers.[1,3]

4 AN EXAMPLE

As an example we want to present a simple dynamical system: The harmonic functions:

$$\begin{aligned} x'(t) &= -y(t) \text{ with } x(0) = 1 \\ y'(t) &= x(t) \text{ with } y(0) = 0 \end{aligned} \tag{1}$$

are to be identified. The well known solution is:$x(t) = \cos(t)$ and $y(t) = -\sin(t)$. As input, we give SCADS measurement of the time behavior of the dynamical system.

In order to give SCADS some prior knowledge we set up a so-called framesystem. The framesystem determines which part of the dynamical system is to be searched, i.e. which subsystem has SCADS to evolve. In figure 2 a subsystem with the name *GP1* has to be found. To asses the quality of the individual, we use the following quality function:

$$Q(x) = \int_0^{\mathrm{T_{max}}} (out_1(t) - y(t))^2 + (out_2(t) - x(t))^2 \,\mathrm{d}t$$

that is strong causal for this identification tasks. Thus we are seeking to minimize this quality.

We tell SCADS our assumption that the result should be an autonomous system.

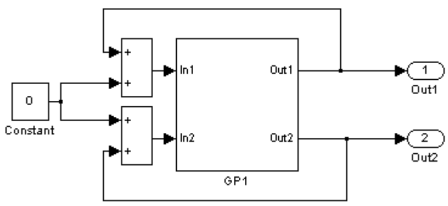

Figure 2: Frame system for evolution of a the harmonic functions.

For the presented simulation, we took a population size of 500 individuals in one population using a partial tournament selection of 25 individual per tournament replacing the 4 worst individuals by 4 newly generated one. We make only 1000 selections then we stop SCADS. This 1000 selections mean, that we have to perform 1000 quality assignments to get the quality of the newly generated individuals and 500 quality assignment in order to evaluate the start population. Figure 3 presents one run of SCADS for the harmonic functions:

We can see, how the minimum values decreases in discrete steps. In this figure the steps appears as straight lines. But in reality the lines has a slight rise because of

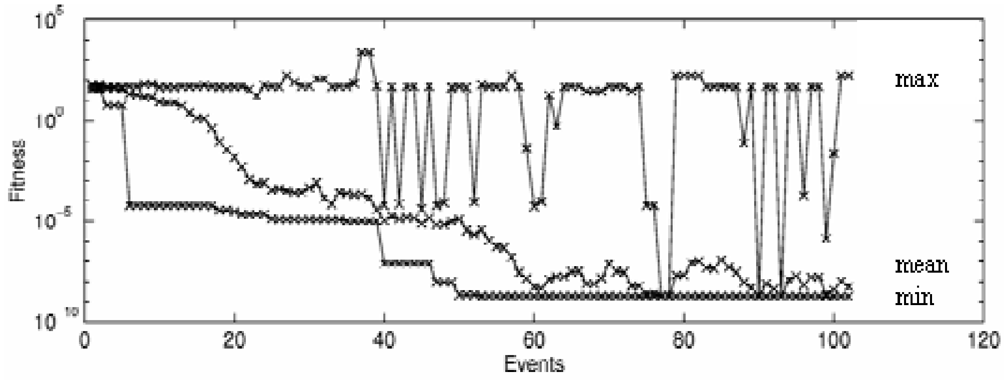

Figure 3: Development of the quality during the search of the harmonic functions. The curves for maximal, minimal and geometric mean value are presented.

the optimization algorithm for the parameters used during the quality assignment. We see also that the mean quality do not change as fast as the minimal quality. This and the fact that maximum quality do not decrease, shows how quality in the population changes, i.e. better individuals become abundant in the population, but during the whole optimization it is possible to generate bad individuals.

Figure 4 shows the best initial system generated by SCADS. The figure shows that the individual is not a connected graph, which contains only transfer functions and simple mathematical operations for the signals. The quality of this system is 20.

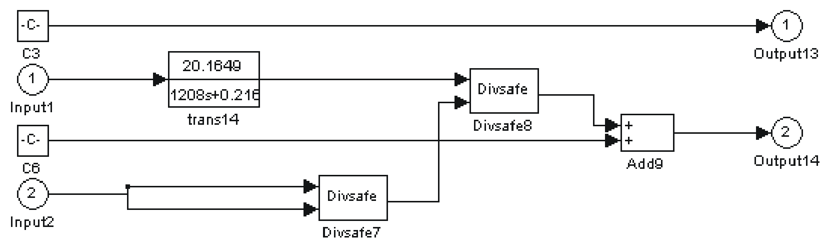

Figure 4: Best initial system for the *GP1* subsystem.

Figure 5 presents the best individual generated by SCADS. Its quality is $1.5 \cdot 10^{-9}$. Thus it is not possible to find any difference to the simulation of the searched system (see equation 1) or to the supplied data. We see also that SCADS generated a full connected graph for the system. One interesting feature is that, due the optimization some values have become large. Looking at the transfer function *transf19* in figure 5, we see that the generated signal is zero. Because of this the integrator with initial value *intc5* can take any initial value, the only prerequisite to get a good quality of the whole system is that the value of the *Gain7* is correlated with the initial value of *intc5* in order to compensate the initial value.

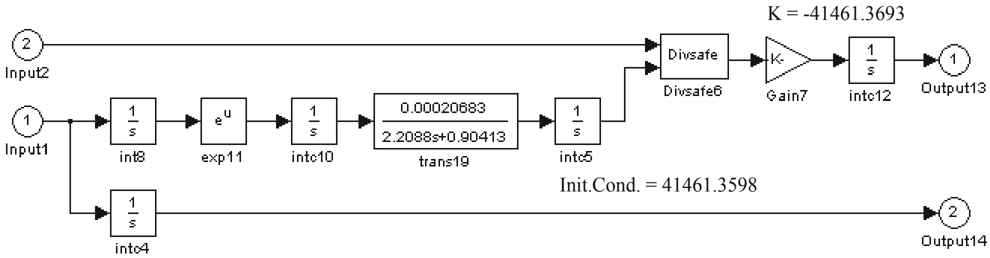

Figure 5: Best final system for the $GP1$ subsystem.

5 SUMMARY AND OUTLOOK

We have presented how the software package SCADS works for two different engineering tasks. We have also demonstrated how is it possible to integrate prior knowledge for the search process. We emphasize the importance of strong causal components in order to achieve an efficient search procedure.

We are planning to extend this tool and to interface not only with MATLAB but also with othe simulation engines such as SPICE.

6 ACKNOWLEDGMENT

The work presented was supported by the grant "GEPROG" from the Federal Ministry of Education and Research and by DaimlerChrysler AG.

REFERENCES

[1] I. Boblan, I. Santibáñez Koref, A. Schütte.*A New Integral Criterion for Parameter Optimization of Dynamic Systems with Evolution Strategy*, VDI Berichte Nr. 1526, pp. 143 - 150, 2000.

[2] R.C. Dorf, R.H. Bishop, *Modern Control Systems*, Prentice Hall, 2000.

[3] S. Hirche, *Strong Causal Quality Functions for automatic Controller Design*, Beiträge zur Bionik und Evolutionstechnik 02/2001, Technical University Berlin, ISSN 1618-8675, Berlin, 2001.

[4] J.R.Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*,MIT Press, 1992.

[5] J.R. Koza et al., *Automatic creation of human-competitive programs and controllers by means of genetic programming*, Genetic Programming and Evolvable Machines, 1 (1 -2) pp. 121 - 164, 2000.

[6] J. Niehaus, W. Banzhaf, *Adaption of operator probabilities in genetic programming*, In Julian F. Miller, et al., ed., Genetic Programming, Proceedings of EuroGP'2001, volume 2038 of LNCS, pp. 325–336, Springer-Verlag, 2000.

[7] I. Rechenberg, *Evolutionsstrategie 1994*, Fromann Holzbog, Stuttgar, 1994.