TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 10/18 pp612-616
Volume 14, Number 5, October 2009

Dynamic Simulations of Nonlinear Multi-Domain Systems Based on Genetic Programming and Bond Graphs*

DI Wenhui (狄文辉)[1,**], SUN Bo (孙 波)[1], XU Lixin (徐立新)[1,2]

1. Department of Computer Science and Technology, Henan Mechanical and Electrical Engineering College, Xinxiang 453002, China;
2. Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

Abstract: A dynamic simulation method for non-linear systems based on genetic programming (GP) and bond graphs (BG) was developed to improve the design of nonlinear multi-domain energy conversion systems. The genetic operators enable the embryo bond graph to evolve towards the target graph according to the fitness function. Better simulation requires analysis of the optimization of the eigenvalue and the filter circuit evolution. The open topological design and space search ability of this method not only gives a more optimized convergence for the operation, but also reduces the generation time for the new circuit graph for the design of nonlinear multi-domain systems.

Key words: genetic programming (GP); bond graph (BG); evolutionary computation; system simulation

Introduction

Optimization of nonlinear multi-domain systems, such as mechanical, electronic, and hydraulic systems, must be based on an initial model structure for the system. There are many methods for such system analyses such as fuzzy theory, neural networks, and operational research methods. However, these methods have problems such as the lack of a unified mathematical model to describe the systems and the limited scope of their applications. In addition, each analysis method has a particular structure which normally leads to structural limitations and is not good at directly expressing knowledge so the identification speed is relatively slow[1]. Analysis of nonlinear multi-domain systems involves the integration of a variety of energy transformation mechanisms, while the majority of system simulation tools are confined to single energy domains, so the system design is very difficult[2,3]. This paper presents a dynamic simulation method based on genetic programming (GP) and bond graphs (BG) to more effectively optimize the design of nonlinear multi-domain systems.

1 Genetic Programming and Bond Graphs

Genetic programming is an extension of the genetic algorithms, which uses the principle of natural evolution to describe the design evolution. Its biggest difference from the genetic algorithm is that genetic programming is an executable program, rather than a string. A frequent objective of genetic programming is to achieve human-type machine intelligence with little direct human involvement[4-6]. The analysis starts with a random population of tree-structured programs related to the design and then uses the genetic operations of reproduction, crossover, and mutation to evolve the

Received: 2009-03-26; revised: 2009-04-28
* Supported by the Basic and Frontier Technology Research Program of Henan Province (No. 082300410390) and Backbone of Young Teachers in University Plan to Subsidize Projects of Henan Province (No. 2005-174)
** To whom correspondence should be addressed.
E-mail: diwenhui@yahoo.com.cn; Tel: 86-373-3691160

program design based on same objective function. Increasingly complex models develop as more information is inserted[7-9]. The program trees that evolve the genetic programming are used in many different ways. Similarly, the genetic programming automatically creates a computer program to solve a problem as the program tree is executed. For example, an algebraic function is generated to approximate a given input-output pattern using standard arithmetic operators and operands. In many cases, the program tree is interpreted as a set of instructions for constructing a complex structure from a very simple embryonic structure. This approach has been used to generate electrical circuits as well as to evolve logic circuits[10]. Program trees are also used to represent modular building blocks, linked by direct lines representing the flow of information. This approach has been used to evolve robust controllers for a given plant design[11]. This article is an example of the second method to achieve the evolution where the program tree is treated as a set of instructions for constructing a complex structure.

The bond graph is a modeling tool which provides a unified method for dynamic system modeling and analysis. The performance of the multi-domain dynamic system can be controlled by unified energy conversion rules which balance the energy input and output to satisfy conservation of energy. With the conservation of energy principle, by connecting an ideal mixture of components, the bond graph model can describe the dynamic behavior of a variety of physical systems. These models can provide very valuable results for dynamic system structures[12].

Bond graphs can be encoded in a genetic programming tree to explore various mechatronic design configurations and parameterizations. The basic bond graph elements are: {*I*, *R*, *C*, Se, Sf, TF, GY, 0, and 1}. A program is initiated by specifying an embryo with test fixtures that are appropriate for the problem. The embryo is an invariant part of a model that contains the interface or boundary information associated with the problem to be solved, such as the drive and the load, or the fixed physical plant. Elements that must be included in the embryo include those defining the system interface at which the desired objectives are measured, since the design performance cannot be measured in their absence[13].

2 Evolution Based on Genetic Programming of Bond Graphs

In the initial stages of the system design, the GP tree represents a bond graph of the embryo procedure. Bond graph embryos usually designate a number of components or bonds which can be modified as the "growing point". The GP procedure starts from these "growing points" to develop into a growth graph through the implementation of a sequence of GP functions applied to the embryonic bond graph[1,13].

The following information is needed before the evolutionary system design begins.

(1) Embryo model (power source, loading, and other parameters for the bond graph) or other sub-models which will not be modified.

(2) The genetic programming operating parameters. Genetic operations, including crossover, mutation, copy, and structure changes, are used to control the generation of an initial population and the evolutionary process to provide an initial population of GP trees to a depth of 3-4 layers.

(3) The fitness function definition (objective function). After each operation on the evolving GP tree, the embryo is bonded to form candidate plans for the design, with the fitness function for the bond graph used to guide the evolution in the desired direction.

At each stage in the evolutionary process, any candidate for the program is subject to a two-step evaluation before being given a fitness value. The whole evolutionary process is driven by the random choice, crossover, and mutation operators. A properly selected fitness function can improve convergence[14] and greatly reduces the time needed for a new generation of the circuit diagram. The evolution terminates when the fitness reaches a specified value with the bond graphs, which meet the fitness constraint given as the bond graph solution for the design[15,16]. The evolutionary process is shown in Fig. 1.

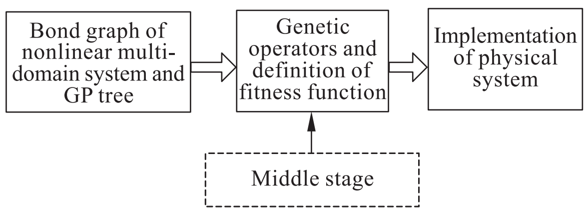

Fig. 1 Evolutionary process