Mechatronics 33 (2016) 71–83

Contents lists available at ScienceDirect

Mechatronics

journal homepage: www.elsevier.com/locate/mechatronics

Design evolution of engineering systems using bond graphs and genetic programming

Buddhika L. Samarakoon*, Lalith B. Gamage, Clarence W. de Silva

Industrial Automation Laboratory, Department of Mechanical Engineering, The University of British Columbia, Vancouver, Canada

ARTICLE INFO

Article history:
Received 9 May 2014
Accepted 20 September 2015
Available online 10 December 2015

Keywords:
Genetic programming
Automated design optimization
Modeling
Mechatronics

ABSTRACT

This paper presents a scheme of evolutionary design optimization, which integrates modeling with bond graphs and optimization using genetic programming for multi-domain engineering systems, particularly mechatronic systems. The performance of the developed system is studied using both experimentation and simulation. During the evolutionary optimization, in addition to the desired response error, system complexity is also taken into account. For the experimental study, the method is implemented in an industrial fish processing machine at the Industrial Automation Laboratory of the University of British Columbia, and the obtained results for suggested design modifications are studied and tested. The drawbacks of the fitness calculation methodologies that are presented in literature are identified and improved fitness functions are developed for evolutionary design in the present work. While previous work has investigated the integration of bond graphs and genetic programming for designing an engineering system, the present work specifically addresses the application of the developed method for the design improvement of an industrial machine. The proposed method is applicable particularly to existing engineering systems, first because the initial model can be tested by comparing its simulated results with the corresponding results from the actual physical system, and second because the design improvements as suggested by the evolutionary design framework, which is developed in the present work, may be implemented and tested against the behavior of the corresponding model.

1. Introduction

Design of mechatronic systems is a complex task that involves different engineering domains such as mechanics, electronics, software engineering and control [1,2]. In designing modern engineering systems, one has to consider all such engineering domains simultaneously in an integrated manner, in order to obtain an optimal design. Furthermore, a design engineer may need to consider other social and economic aspects during the design process. This presents a vast design space to the design engineer which cannot be explored manually in a systematic and convenient manner. In this context mechatronic system design can be formulated as a problem of multi-objective search and optimization [3,4].

Genetic Programming (GP) is a well-known method of stochastic search and optimization that can be used in multi-domain, mechatronic design problems [5,6]. Inspired by biological evolutions, GP considers a set of potential solutions at a time. By carrying out genetic operations, subsequent populations are obtained until a desired solution is achieved. In genetic programming, each solution is represented as a tree structure. Initially, GP was used for automated program creation by Koza [7]. GP was used to handle the aspects of system design in early stages of a design (conceptual design). Kristinssen and Dumont employed GP for identifying poles, zeros and gain of a controller in this context [8]. In the work of Grimbleby et al. [9,10], genetic programming for synthesizing electrical circuits has been proposed. In their work a modeling method to represent the solutions was not used and that restricted the applicability of the methods in different engineering domains.

Initially proposed by the celebrated engineering professor Henry Paynter of MIT [11], bond graph (BG) modeling is a technique that represents the power flow between components within a system. As power is a physical quantity that is common to any engineering domain, bond graphs were extensively used for modeling multi-domain systems [12,13]. This method, combined with GP made evolutionary design of multi-domain engineering systems possible. Furthermore, BG provided a convenient tool for simulation and evaluation of identified solutions. Moreover, by using bond graphs, one can conveniently choose the state variables to build mathematical models and to convert them into signal flow

* Corresponding author. Tel.: +1 305 877 3569.
E-mail address: buddhika@interchange.ubc.ca (B.L. Samarakoon).

http://dx.doi.org/10.1016/j.mechatronics.2015.09.009
0957-4158/

graph [14]. In the work of Tay et al. [13] Genetic Algorithms were used for the design modification of an air pump. Because of the limited number of solutions that can be represented in Genetic Algorithms, it cannot be applied for a design problem with a significantly large solution space. Fan et al. [4] initially introduced genetic programming for multi-domain engineering design. Their work demonstrates the identification of electrical filters and micro-electromechanical systems (MEMS). However they have not addressed problems such as complexity of the evolved systems. For example, the high pass filter evolved in [4] has unnecessary energy storing elements, without any specific relation to the transition bands. Recently Behbahani and de Silva proposed hybrid evolutionary algorithm [15] which essentially separates the design search space into two domains consisting of topology and parameter values. Simultaneously optimizing both topology and sizing may not work in a complex design spaces. This was also observed in our previous work [16,17] and we have proposed to include a fitness criterion based on system order, which will limit the topological design space. As presented in [15], this paper also analyzed a case where a system having proper topology obtains a lower fitness value compared to a system with incorrect topology. We propose a better fitness criterion in order to overcome that.

Even though many researchers have improved this BG-GP based methodology [18,19] the work presented in this paper focuses on the development of an evolutionary design framework for an industrial machine. This work builds upon the design evolution framework proposed by Gamage and de Silva [20]. In this framework, the industrial machine is connected to a Machine Health Monitoring System (MHMS), which is capable of detecting malfunctions [20–25] and identifying faults of the system. This paper considers the malfunctions due to design faults and assumes that the faulty subsystem can be isolated with the assistance of a design expert system. The faulty subsystem is modeled using bond graphs and an improved design is evolved using genetic programming. The Bond Graph models should not be allowed to freely evolve since some of the evolved solutions may not be feasible for actual implementation [16,26]. Another aspect that has not been previously considered is, BG-GP methodologies have not been used for an actual industrial system. The framework that is presented here demonstrates the application of BG-GP methodology for design improvement of an industrial fish processing machine called “The Iron Butcher” [27].

The work presented in [21,22] is focused on condition monitoring of an industrial machine. Razavi et al. [22] modeled a hydraulic system using a UKF approach. The estimated states and actual system values were compared to identify any existing faults. Raman and de Silva [21] developed a set of features from accelerometers and microphones which were used as the inputs for a neural network or support vector machine based classifiers. While their work fits with the overall framework presented in Gamage and de Silva [20] our work differs by proposing design improvements once a fault has been identified. This is achieved using a iterative process based genetic programming. Furthermore, this work assumes the faults identified by methods in [21,22] has been used as an input for the work presented in this paper. Compared to the work done by Gamage et al. [23] the present work uses Bond Graph as the modeling tool because of aforementioned advantages. Use of bond graphs helps in identifying the independent states in the industrial machine presented in Experimental Validation section. This is further explained in Section 4.2.

2. Methodology

There are two main functions in a multi-domain engineering design procedure. They are obtaining optimal connections between the components and obtaining proper parameter values of the components [9,23]. As an example, in an analog filter design problem, the same set of resistors, inductors and capacitors can be organized to a circuit in numerous ways. Out of these possibilities, the design engineer must choose the set of connections that gives the required performance and also the corresponding numerical values for the components (capacitance, resistance, etc.). The two main parts of this evolutionary process are, representation of the engineering system using bond graph modeling and the use of genetic programming to evolve an optimal design.

2.1. Solution representation for the evolution

Since GP employs tree structures for the evolutionary process, the multi-domain engineering system has to be represented as a tree structure. In a BG-GP based approach, the bond graph model of an engineering system is constructed as a tree [4,28]. The evolutionary process will be started by first having an embryo model of the system. The embryo contains the information about the basic topology, inputs and outputs to the system and the locations of the system that are modified during the evolutionary process. An embryo and its bond graph representation are illustrated in Fig. 1 .

The basic elements used for modeling the systems in this work are capacitors, inductors, resistors, voltage sources, current sources, transformers and gyrators and the mechanical counterparts of these elements [2]. During the evolution, the embryo is evolved until an optimal solution is obtained. The modifications include, changing the parameter values of components, adding or changing of components and changing the topology of the system. These modifications are carried out through construction functions. The construction functions used in this work are presented in Table 1. The chosen construction functions include arithmetic operations, which are required for modifying parameter values and also they can modify the topology and the components of a system. As in all the evolutionary schemes, GP also considers a set of candidate solutions, which is called a population, during the optimization [26]. Given the embryo model, the initial

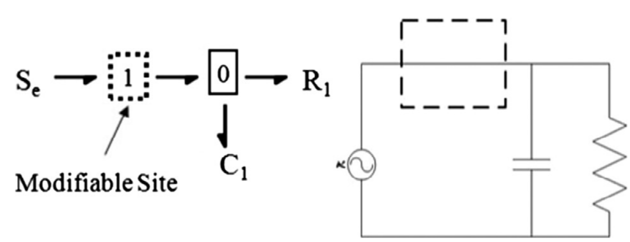

Fig. 1. Embryo model.

Table 1
Construction function set.

ID	Construction function
21	Adding a resistive element to a junction
22	Adding a inductive element to a junction
23	Adding a capacitive element to a junction
20	End of operations
30	Inserting a common effort junction to a bond
31	Inserting a common flow junction to a bond
12	Adding two numerical values
13	Subtracting two numerical values
14	Multiplying two numerical values
15	Dividing two numerical values
16	Exponentiation
17	*N*th root
11	Numerical value for arithmetic operations