Mechatronic Design Evolution Using Bond Graphs and Hybrid Genetic Algorithm With Genetic Programming

Saeed Behbahani and Clarence W. de Silva, *Fellow, IEEE*

***Abstract*—A typical mechatronic problem (modeling, identification, and design) entails finding the best system topology as well as the associated parameter values. The solution requires concurrent and integrated methodologies and tools based on the latest theories. The experience on natural evolution of an engineering system indicates that the system topology evolves at a much slower rate than the parametric values. This paper proposes a two-loop evolutionary tool, using a hybrid of genetic algorithm (GA) and genetic programming (GP) for design optimization of a mechatronic system. Specifically, GP is used for topology optimization, while GA is responsible for finding the elite solution within each topology proposed by GP. A memory feature is incorporated with the GP process to avoid the generation of repeated topologies, a common drawback of GP topology exploration. The synergic integration of GA with GP, along with the memory feature, provides a powerful search ability, which has been integrated with bond graphs (BG) for mechatronic model exploration. The software developed using this approach provides a unified tool for concurrent, integrated, and autonomous topological realization of a mechatronic problem. It finds the best solution (topology and parameters) starting from an abstract statement of the problem. It is able to carry out the process of system configuration realization, which is normally performed by human experts. The performance of the software tool is validated by applying it to mechatronic design problems.**

***Index Terms*—Bond graphs, electrohydraulic systems, genetic algorithms, genetic programming.**

I. Introduction

MECHATRONICS refers to the field of modern systems and technologies, where a synergic integration of mechanical, electrical, computer, and information technologies is used for improving control, automation, efficiency, and intelligence [1]–[3]. Due to its multidomain nature and associated dynamic interactions, solution to a mechatronic problem should take an integrated and concurrent approach [4]–[10]. Also, since the design space becomes very vast, due to the existence of many possible combinations of solutions, the problem solution needs creativity and expertise, especially in terms of the realization of the system configuration.

An engineering design entails several phases. Conceptual design is the first phase, where high level decisions about the combination of subsystems and the required function of each subsystem are made. The high level description and the expectation of each subsystem are specified, while detailed system configuration and the associated design calculations are not performed. Later, in the detailed design of each subsystem, first the best topology for each subsystem must be realized. To optimally design a mechatronic system, we first need to realize the best topology for the system. Topology is the number of different elements used in the system, and the way they are interconnected [11]. Later, the inherited parametric values of the elements of the particular topology have to be specified. When the topology is realized, and when the design space only entails numerical parameters, different optimization tools and algorithms may be utilized to optimally find the unknown parameter values. The mechatronic problem is usually complex because the topology of the best solution is not necessarily obvious. Since determination of the topology of a system is an open-ended problem, it creates a potentially extensive search space for the designer. Due to this complexity, topology design is usually viewed as a problem which needs human intelligence and should be performed by experts.

There have been efforts in the past to create tools to automatically generate the topology and size of the solution to a mechatronic problem. Hu *et al.* [12] addressed some important characteristics of topology search, which are common in all system synthesis problems (e.g., electric circuit and controller synthesis), including low neutrality, epistasis, and high multimodality of search space, discreteness, and the lack of information to guide local topology search. In fact, due to these characteristics, the past efforts in automating the system synthesis have primarily employed evolutionary algorithms. Seo *et al.* [11], [13] introduced the idea of integration of bond graph modeling and genetic programming with the aim to synthesize the topology and sizing of a mechatronic system. In [12], the eigenvalue placement problem has been proposed as a scalable benchmark problem to investigate the abilities of computational synthesis tools. Also, the effectiveness of the integration of BG and GP for this topology benchmark problem was elaborated. However, the idea of using genetic programming for topology realization had already been utilized by Koza for the automated

Manuscript received February 3, 2011; revised May 9, 2011; accepted July 19, 2011. Date of publication September 26, 2011; date of current version September 12, 2012. Recommended by Technical Editor O. Kaynak. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, in part by the Canada Research Chair in Mechatronics and Industrial Automation, in part by the Canada Foundation for Innovation, and in part by Isfahan University of Technology (IUT), Iran.

S. Behbahani is with the Mechanical Engineering Department, Isfahan University of Technology, Isfahan, 84156-83111, Iran.

C. W. de Silva is with the Mechanical Engineering Department, University of British Columbia, Vancouver, BCV6T 1Z4, Canada (e-mail: desilva@mech.ubc.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2011.2165958

design of electrical circuits [14], [15]. Koza utilized a combination of GP with a graphic representation of electrical circuits, while Seo *et al.,* used BG as graphic representation of design individuals in the course of GP optimization. They both introduced several construction functions and terminals to be applied to an embryo model, to generate a variety of design individuals. After Seo's work, Wang *et al.* [16], [17] extended the integration of BG and GP for the automated design of the controller subsystem of a mechatronic system. They showed that we could acquire useful technical knowledge about control design from the results of this algorithm. BG is normally used to model the power domain of a mechatronic system, while the controller belongs to information domain. To represent the controller in BG similar to the power domain part, the method of "controller design in the physical domain" was applied in [16], where a linear controller was represented by an equivalent combination of basic lumped elements (i.e., capacitance, inductance, and resistance). In our previous work, we extended the outlined idea for nonlinear systems, and in particular we developed several nonlinear construction functions to utilize the generated tool for black box nonlinear system identification [3], [18].

There is considerable literature on the application of evolutionary algorithms in the automatic synthesis of the shape of a mechatronic structure—a problem that has important similarities to the topology synthesis, which is addressed in the present paper. Grossard *et al.* [19] developed an evolutionary tool, called Flexin, to automatically synthesize a truss-like planar structure made of passive and active building blocks, made of piezoelectric material. The authors produced a library of compliant building blocks, which are combined in the course of the evolution, to approach an optimum solution, through calculation of the fitness of the generated individuals. They applied a multicriteria fitness evaluation to address both conventional mechanical criteria and control criteria concurrently. Zhang *et al.* [20] designed the shape of the fins of a swimming microrobot, manufactured using magnetostrictive thin films, through a genetic algorithm.

The ideal objective of an evolutionary algorithm is to mimic natural evolution which happens at very slow speeds. In the engineering world, evolution of an industrial system from its initial creative solution to a modern system happens gradually and continuously over the years, and is led by human experts. The present paper outlines the specifications of natural evolution for an industrial system, and demonstrates that the design model outlined in prior work does not exactly coincide with these specifications and has some drawbacks. The present paper asserts that most drawbacks of the model are due to its deviation from natural evolution. Then, the paper proposes a new design model, by considering a two-loop optimization using a hybrid of GA and GP integrated with BG. The paper justifies that the new model agrees better with our understanding of evolution, and provides significant improvement over the earlier model.

The proposed design model has been used to develop a unified evolutionary mechatronic design tool. The mechatronic design problem is expressed as an optimization problem, along with a fitness evaluation program. For comparative evaluation of the capabilities of the developed tool, several mechatronic design problems that are available in literature are chosen. The obtained results verify that the proposed model provides better and more trustworthy designs; however, the design process is more time consuming.

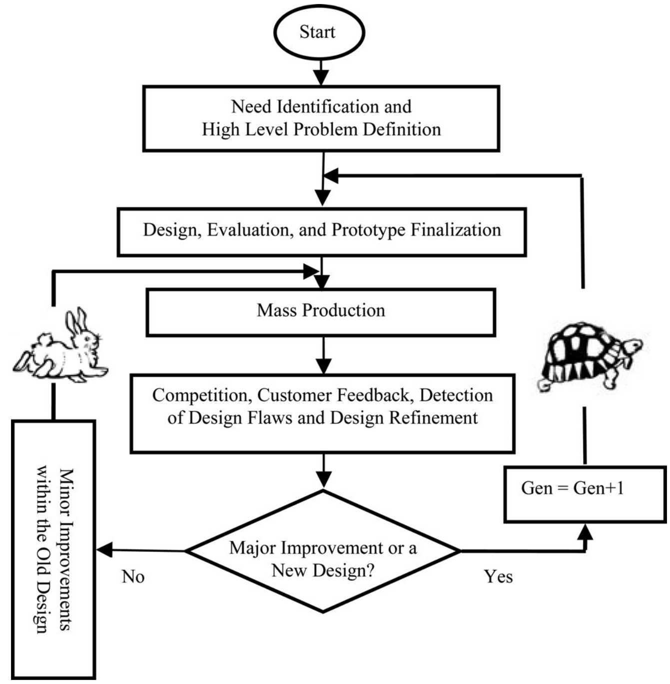

Fig. 1. Flowchart of natural evolution of an engineering system.

The rest of the paper is organized as follows. The next section presents the proposed two-loop evolutionary design model and justifies that this model has a better fit with natural evolution. Section III explains the developed tool. First, a brief introduction to the basic elements of the tool—BG and GP—and also the earlier work in the integration of BG and GP is presented. In section IV, several case studies are presented to demonstrate the capabilities of the developed tool.

II. Two-Loop Dsign Model

Consider a typical engineering system or machine that we encounter in our activities. Most likely, it has undergone significant improvement from its original design. In its evolution history, one may be able to identify several generations where significant topological improvement has taken place in comparison to the previous generation. In each generation, the system structure and the topology might be quite similar. Even after considerable design improvement, the primary high level description of the system might be the same, while the detailed design might be significantly different with regard to the structure and the topology. This process of evolution of an engineering system is represented in Fig. 1. It is seen that a two-loop optimization exists, where the rate of repetition inner loop is quite faster than that of the outer loop. The inner loop represents the minor, insignificant, and nontopological changes in the designs of a generation (i.e., just numerical changes), while the outer loop shows the topological and structural improvements (i.e., topology realization).

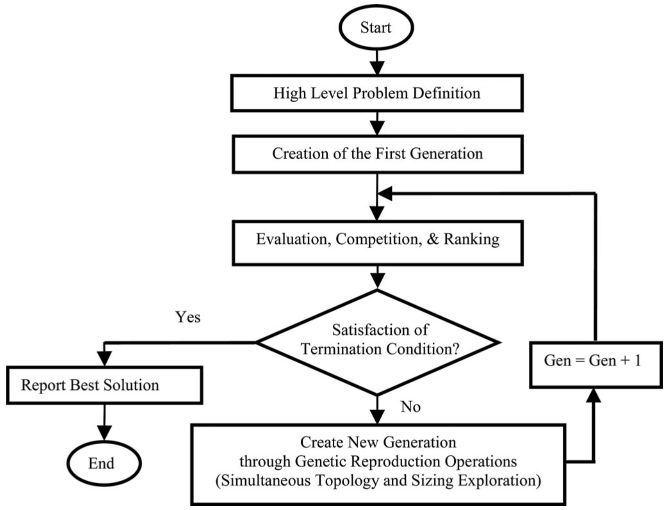

Fig. 2. Early design model for system synthesis.

The process of topology realization requires considerable human expertise, intelligence, and creativity. Efforts have been made to make this process somewhat autonomous. In particular, attempts have been made to develop a tool that is able to develop a system structure from a high-level problem statement [11]–[14], [18]. A structure entails both topology, and sizing of the solution. The process of topology realization and sizing optimization were carried out simultaneously, using genetic programming. An associated design model is shown in Fig. 2.

This model has the following drawbacks:

1) It does not agree with the natural evolution as presented in Fig. 1. It attempts to achieve best topology and sizing simultaneously (i.e., simultaneous minor and major modifications) in a random search. It is rather optimistic for a tool to optimize the topology and sizing simultaneously.
2) The topology realization is a very complex problem with a vast search space. The most important and vital part of the optimization is to find the best topology. A genetic algorithm is a random search method, which involves a competition for survival between randomly generated solutions. In this process, solutions with lower fitness values are most likely removed and not allowed to participate in creation of the next generation. Suppose that the optimum topology is achieved through GP, but its sizing is not appropriate; hence, its fitness would be low. On the other hand, a solution may have a rather high fitness, while its topology is far from optimal. This will mislead the optimization process.
3) The optimum topology may lose in the competition, and may not be allowed to participation in the GP process. This is not acceptable in the topology realization, and may eventually result to a nonoptimal solution. To remedy this situation, we should let any generated topology to evolve under its own topology, and let the *elite* representative of that topology to compete for survival. Then the decision to keep a topology or allow an individual to participate in the next generation creation will become more rational.
4) Another concept in natural evolution of species and in biologically inspired genetic algorithms is the concept of niching genetic algorithm. Here, individuals compete with each other over resources only if they have some similarities. Under this concept, several colonies can exist without any competition among the colonies, and the members of one colony can be much weaker than those of another colony. However, there will be continuous competition between individuals in a given colony to survive and to become the elite of that colony. It is observed that this important feature and the survival of species are mimicked in the suggested design model. As a result, the final outcome of the optimization is not just a single best solution but a set of colonies (different topologies) with a representative elite for each colony (the best size for a particular topology). This is quite useful in mechatronic design due to two reasons. First, in many engineering problems, we need to present different optimal solutions, so that the user can select between them based on his/her own preferences and limitations. Second, in engineering design, there will always be some design criteria that are not represented in the fitness function evaluation. Handling some of them may be very time consuming, and some may need referral to manufacturer's catalogs, which are typically not accessible in the program loop of optimization. In this case we may have different elites, and then allow them to participate in a final competition, where more design criteria and more dedicated evaluation are considered.

In this work, a two-loop design model is proposed, quite similar to the natural evolution of engineering systems. The flowchart of this model is given in Fig. 3. In this model, individuals first compete with solutions, which are topologically similar to themselves, to represent the elite of the current topology (i.e., the inner loop). Since the topology is maintained similar, only the numerical values of the parameters are the subject of the competition and optimization. This competition and optimization is achieved by means of a conventional genetic algorithm. When the elites of different topologies are found, their elites compete and participate in the genetic operations (i.e., crossover and mutation) with other elites to create the next generation, and so on, until the global optimum is attained (i.e., the outer loop). Represented by their elites, the topology of the individuals is the subject of the competition and evolution in the outer loop, which is accomplished by GP. Four main drawbacks of the earlier model were addressed earlier in this section. By reviewing these drawbacks it can be observed that the proposed two-loop design model and the associated hybrid of GA and GP for implementation of the proposed model will efficiently overcome the mentioned drawbacks.

III. Mechatronic System Synthesis Tool

In the implementation of the proposed design model, an integration of bond graph modeling as a tool for integrated mechatronic modeling with genetic programming is utilized for

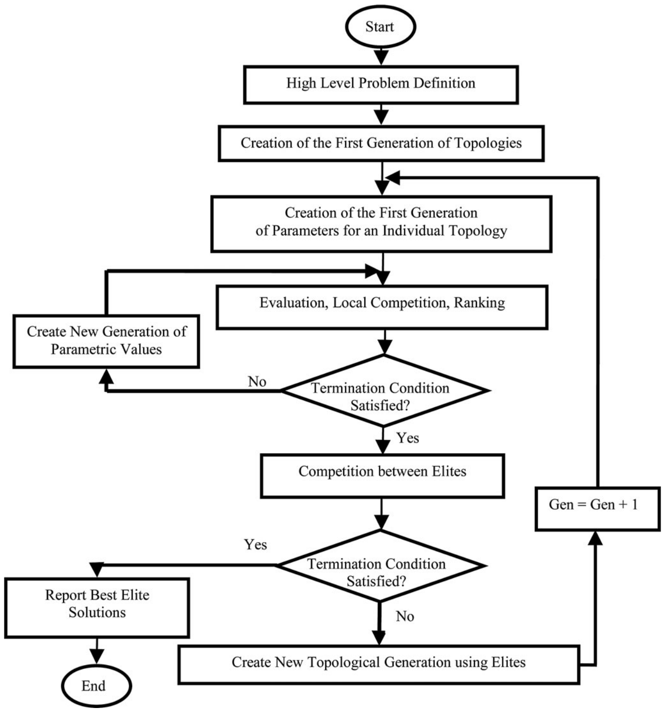

Fig. 3. Two-loop design model for mechatronic system synthesis.

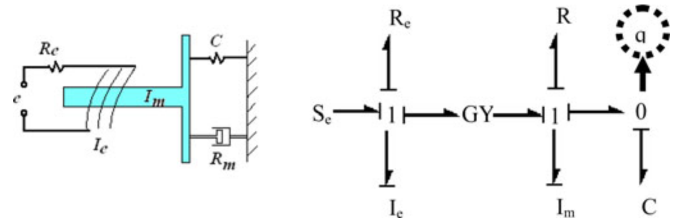

Fig. 4. Schematic diagram and the bond graph model of the torque motor of a servo valve.

The present paper develops a tool for automated synthesis of the optimum model of a mechatronic problem. In the course of optimization, different design individuals are generated in BG representation. These models have to be evaluated against the desired behavior of the target system. The evaluation usually requires simulation of the model. To evaluate the generated BG models, a BG modeling and simulation tool is developed. A new matrix-based formulation is proposed and used in the development of this software tool [2]. The final output of the tool includes a set of state space equations for the designed model, which can be used for the simulation and evaluation of the designed model.

topology exploration and optimization (outer loop). This idea is inspired by the earlier work by Seo [11] and subsequent extensions presented in several papers [12], [13]. A framework is presented, where a tree-like structure of construction functions is mapped to a bond graph model. The tree-like structure of each individual model has all the required characteristics for representation of solutions for genetic algorithms (i.e., branches are like genes, and crossover and mutation are applicable). As the inner loop, and for realization of the elite of each topology, a simple genetic algorithm is utilized [21]. Once a topology is generated in the course of the GP process, the number of required numeric parameters is identified. Then, a string of binary bits is used to represent an individual set.

In the next section, basics and characteristics of bond graph modeling and genetic programming are outlined. Then, a framework for integration of these elements in the context of the proposed two-loop design model is presented. The main differences between this design optimization framework and the framework used in the previous work are explained.

A. Bond Graphs (BGs)

Mechatronic systems need an integrated multidomain modeling environment, where different subsystems such as mechanical, electrical, hydraulic, and control may be modeled by the same approach. In this context, bond graphs (BG) provide an effective and accepted method [22], [23]. BG has a domain-independent language and a graphical representation that can be used to model basic elements and connections in different fields. As an example, the BG model of a torque motor of an electro-hydraulic servo valve is shown in Fig. 4 [3].

B. Genetic Programming

Evolutionary algorithms provide significant advantages for optimization in some types of problems as considered in the present work. They effectively handle complicated or ill-modeled problems, where an explicit formulation for the evaluation of the effect of the variation of design parameters on the cost function cannot be developed. Also, unlike the gradient-based optimization methods, continuity and differentiability of the cost function are not key requirements [24].

All branches of evolutionary algorithms, including genetic programming (GP) and genetic algorithms, try to mimic the natural evolution of biological species. They have a chromosome-like representation of individuals, where changes in the genes can result in changes in the behavior and properness of the individual solution [24], [25]. GP differs from the conventional GA with respect to the particular chromosome-like representation of the individuals. In the conventional GA, a series of zeros and ones is used in the representation, as shown in Fig. 5(a). In contrast, GP utilizes a tree-like structure for the representation [Fig. 5(b)] [13]–[18], [26]. The branches of the tree-like structure behave similar to the genes. Any change in the branches of the tree-like structure, or switching a branch with a random branch, or exchanging branches between two different tree-like structures will result in new structures, with some characteristics inherited from the structure that generates it.

C. Integration of BG and GP

By adding different random combinations of bonds and nodes to modifiable sites of an embryo, different new models with different topologies can be created. Also, switching some branches of two individual models will result in two new models. These operations are basic requirements for developing evolutionary

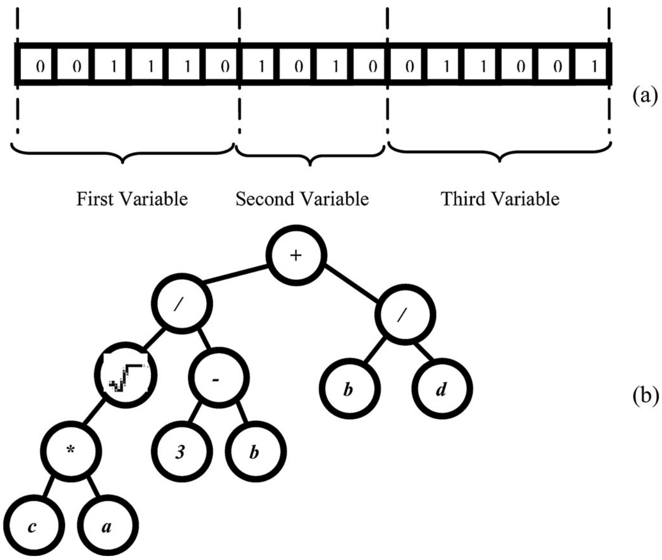

Fig. 5. Chromosome-like representation of individual solutions: (a) Conventional GA. (b) Tree-like representation in GP.

algorithms, which have been utilized in the present work and also some earlier work [11]–[13]. Each tree-like representation of the composed construction functions will result in an individual model. After evaluating all the models in a generation, crossover and mutation operations are utilized based on the fitness of the individuals to explore the search space. The randomness of these operations along with the open-ended nature of the BG representation, results in an open-ended topology exploration scheme. This will lead to an optimum design consisting of an optimum combination of the topology and the parametric values for a particular mechatronic problem.

Each tree-like structure of the construction functions should map to a BG model. Two main elements of this mapping are the embryo model and the construction functions which are explained next. Other elements like fitness evaluation, selection method, and genetic operations are rather common in any genetic algorithm. Further explanations of these concepts can be found in [24].

D. Bond Graph Embryo

The branches of a tree-like structure include construction functions, which will be eventually applied to an embryo model to generate a trial design. The embryo model, which has to be given by the user for any particular problem, includes basic and abstract specifications of the system to be designed. It will represent the common part of all generated trial design individuals. In compliance with the modification and extension requirements in the topology of the system, the embryo model has *modifiable sites* and *open sites* [11].

A design problem still needs human knowledge and intelligence in generating the embryo model, and also in reflecting the system requirements in the definition of the fitness function. However, the required knowledge and intelligence is much lower in comparison with the case when the topology realization is performed by humans. The developed tool will succeed

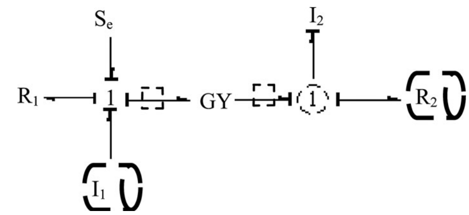

Fig. 6. Sample BG embryo model.

as long as the given embryo is correct and specifies the input and output ports of the system.

The present work utilizes almost the same framework proposed initially by Seo *et al.* [11]. A typical embryo model is shown in Fig. 6. Three types of modifiable sites are observed:

1) Modifiable joints, shown by dashed circles.
2) Modifiable bonds, shown by dashed squares.
3) Arithmetic sites, shown by dashed cylinders.

Modifiable joints shown by dashed circles allow for extension of the embryo by adding new elements to the nodes. Modifiable bonds shown by dashed squares allow extension of the connections between nodes. Arithmetic sites shown by dashed cylinders correspond to modifiable numerical values of the elements in the embryo and also the added elements. Arithmetic sites are necessary for the sizing optimization of a created model.

E. Construction Functions and Terminals

Construction functions add an element, or make a change in the model, when they appear in a tree-like structure. Terminal refers to the location in the model where the modification of the construction function is performed. Seo *et al.* [11] introduced several construction functions for the integration of BG and GP. In their framework, the embryo model and a model generated from it can have three types of modifiable sites. Each type can accept its own available construction functions. For each type, an *End Function* exists among available functions. This is because other construction functions generate new modifiable sites and also keep old modifiable sites for further improvement. This feature is the core concept of the integration of BG and GP for topology exploration. It allows the growth of the embryo to different BG models. If an *End Function* is not included among the available options, the growth will be unlimited.

For a modifiable node, a lumped 1-port element (i.e., resistance, capacitance, or inductance) may be added, or it may be considered without any extension. This is decided randomly, upon the generation of a random number and specification of predefined ranges for the selection of each option. When a new element is added, two new modifiable sites are created and the old modifiable site is also kept for further modification (Fig. 7a). It also has an arithmetic modifiable site, which corresponds to the value of the added element.

For a modifiable bond, a junction (i.e., 1-junction or a 0-junction) may be added, or alternatively it may be decided that the site is left without any extension. Again, this is decided randomly. If an Add-junction function is applied, it will have three arguments or create three new modifiable sites. Two of

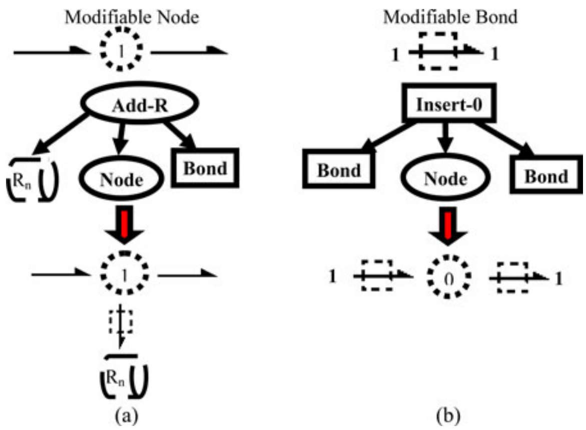

Fig. 7. Construction functions: (a) Add-R function applied to a modifiable node site. (b) Insert-0 function applied to a modifiable bond.

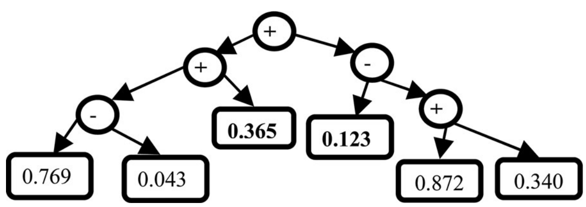

Fig. 8. Example of the tree-like representation of an arithmetic modifiable site.

them are bond-type and the third one is node-type, as shown in Fig. 7(b).

In previous work, a tree-like genetic representation (see Fig. 8) is considered for an arithmetic site as well, which would result in a numerical value for a lumped element. In addition, the parametric value realization is treated in the same level of topology realization. In other words, the tree-like representation for any individual solution contains branches for structure generation and branches for numerical optimization. The optimization procedure was based on the model presented in Fig. 2 – its drawbacks were explained in Section II.

In the present work, a two-loop optimization method is followed. It is based on the design model shown in Fig. 3. Once a topology is suggested in the course of GP exploration, the number of required parameters for that topology is determined. Subsequently, an optimization process using GA is performed to find an elite representative for this topology (see Fig. 9).

GA is implemented and tuned such that unless an individual topology is generated purely in random manner (e.g., first iteration in the outer loop) or a particular element is placed purely randomly in an evolved topology (i.e., through mutation), the GA continues through the previous evolution to improve the parametric values of the particular topology. The parametric values from the previous evolution are inherited by the generated topology, and GA is employed to approach the optimum correction factors—instead of the parametric values themselves—to improve the fitness. Also, a no-correction individual is placed in the first generation of GA (i.e., random generation) as a candidate, to ensure that the fitness is not reduced. A feasible value for each type of element is considered as a suggestion, if no

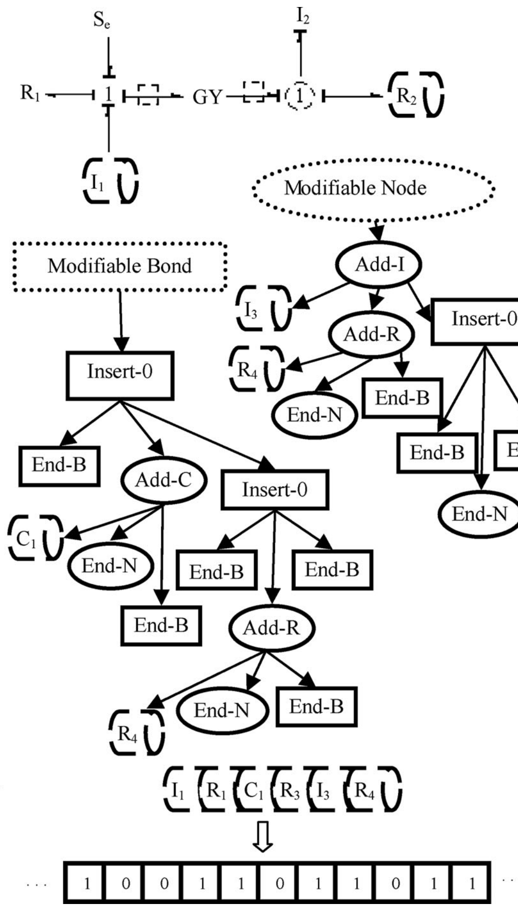

Fig. 9. Creation of a string to be used in the GA-based realization of an elite for an individual topology.

previous evolution exists. With this strategy, one does not need to overly worry about the population size and the number of generations in the inner loop (i.e., GA). The population size and the maximum number of generations are set to 30; however, the attempt is terminated if no significant improvement is observed in four succeeding generations.

In the outer loop, the best elite of each topology will represent this topology for competition with other topologies, and also for creation of the next generation. The advantages and justification of this design model were explained in Section II.

IV. Case Studies

In this section, case studies are presented to demonstrate the abilities of the developed tool. The objective of these case studies is to stress on the topology realization abilities of the developed tool. The first example includes the black box design of passive low-pass and high-pass analog filters, which have been studied in the previous work as well [11], [14]. The next example concerns autonomous synthesis of a controller for a highly nonlinear electrohydraulic system called Iron Butcher—an intelligent mechatronic system which is designed and built for the fish processing industry to automatically cut the head of fish with minimum meat wastage.

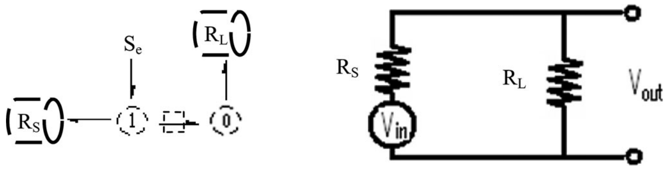

Fig. 10. Embryo model for analog filter design.

A. Analog Filter Design

Analog filter design is a symbolic problem, which is used for evaluation of the system design tools. For a low-pass filter, the objective is to design a system that passes the low frequency components of a signal without any change, and completely removes the high frequency components. It means that the ideal system has a unity transfer function value up to a cutoff frequency and a zero transfer function value after that. The ideal response of a high-pass filter is the opposite of that of a low-pass filter, i.e., zero transfer function value up to the cutoff frequency and unity after that. In the following examples, the cutoff frequency of 1000 Hz is used. These characteristic descriptions are high level and abstract statements pertaining to the circuits to be designed. The purpose of the developed tool is to synthesize an appropriate electric circuit to meet each of these abstract design statements. The embryo model used for these designs is shown in Fig. 10.

The ideal responses of the systems are used to define the fitness functions, as follows:

$$F = \sum_{w_i=1}^{10^{2.4}} (H_i - H_d)^2 + \sum_{w_i=10^{2.4}}^{10^3} (\log(w_i) - 1.4)^3 (H_i - H_d)^2 + \sum_{w_i=10^{3.09}}^{10^{3.6}} (4.6 - \log(w_i))^3 (H_i - H_d)^2 + \sum_{w_i=10^{3.6}}^{10^6} (H_i - H_d)^2 \quad (1)$$

where H_i represents the attenuation coefficient of the filter at a particular frequency, while H_d is the ideal attenuation coefficient. This function consists of four terms for four different regions of the frequency band. It has been defined in such a way as to give more power to the points, which are close to the cutoff frequency. Twenty points were considered for each of the first and the fourth terms, while 40 points were considered for the second and the third terms. Also, the second and the fourth term have an additional term, which increases the role of the points closer to the cutoff frequency. However, a small range of $10^3 - 10^{3.09}$ Hz is not considered in this evaluation. The evolved circuits and their frequency response are shown in Figs. 11 and 12. It is clear that the evolutionary tool has been able to find appropriate circuits for both cases, with a very low number of components in comparison with the results presented in the earlier work [11], [14]. It can be justified by the fact that each topology has had a chance to represent its elite for competition with other topologies. The possibility of that optimum topology losing the competition is very low; hence, the algorithm will achieve the global optimum at high probability.

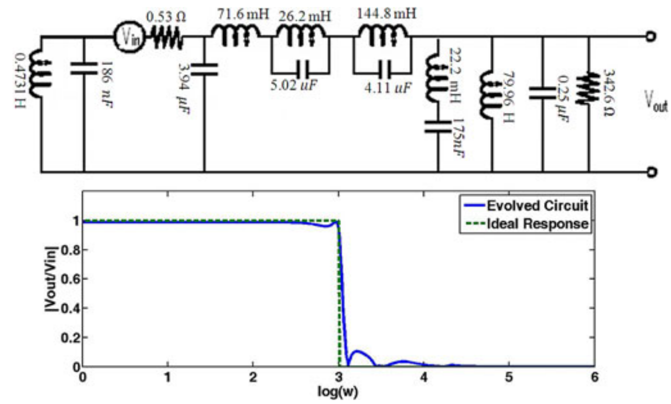

Fig. 11. Evolved low-pass filter and its FRF.

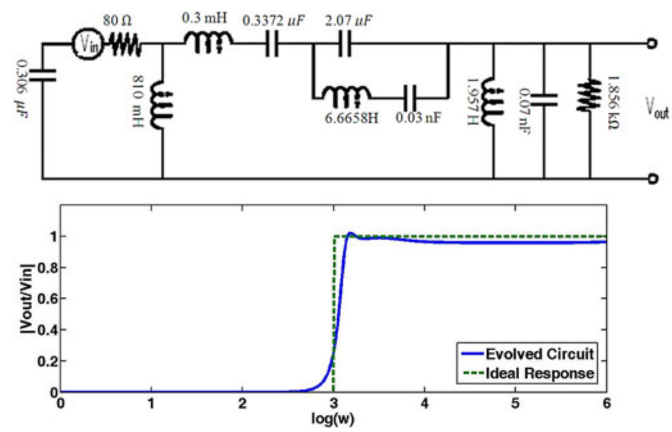

Fig. 12. Evolved high-pass filter and its FRF.

B. Iron Butcher Controller Design

The Iron Butcher is an automated industrial machine, which has been designed and developed in the Industrial Automation Laboratory of UBC for head removal of fish with minimum meat wastage [27]. In this machine, a vision system is used to observe the gill of a fish, from which the best cutting location for the fish is determined. The corresponding coordinates are sent to a two-degree-of-freedom positioning table to optimally position the cutter. Two electrohydraulic manipulators with two servo-valves are used in the positioning table (Fig. 13).

In the present work, the developed tool is utilized to automatically synthesize an appropriate controller for this highly nonlinear machine, starting with a high level statement for the desired behavior. In an earlier work, the BG model of the electrohydraulic system has been developed [3]. In the present work, since this model has to be simulated many times during the course of the optimization, a simplified model is utilized.

The block diagram of the system is shown in Fig. 14. The input to the system is the current to the servo valve. The measured output of the system is the position. Also, pressures at both sides of the piston are measured, which are used for calculation of the force exerted on the piston. Friction is present as a disturbance

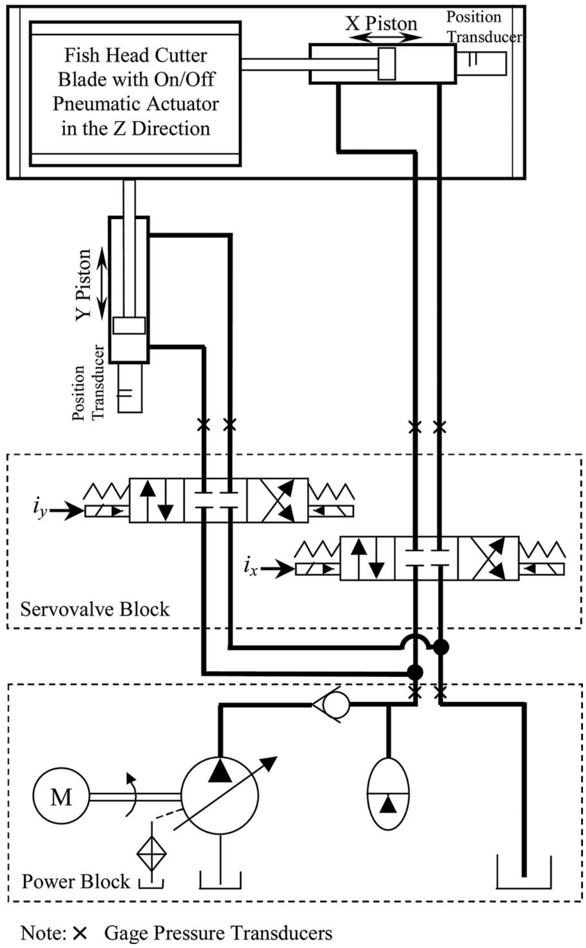

Fig. 13. Schematic diagram of the electrohydraulic manipulator.

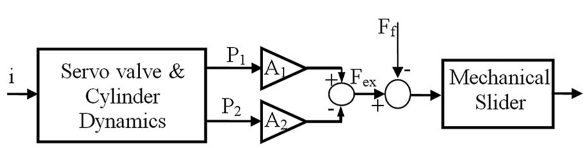

Fig. 14. Block diagram of the original system.

to the system and is not measured. It highly affects the dynamic behavior of the system. Since a main challenge in controlling the system is friction, a friction observer together with a velocity observer has been added to the system, so that the estimated friction and the velocity are also available to the controller in each time step. The block diagram of the observers is given in Fig. 15. The velocity observer is designed using pole placement. The poles of the observer are placed at -200 and -300 to guarantee sufficiently fast convergence of the observation in comparison to the controller speed.

It is known that surface contact friction is a nonlinear function of the relative velocity and its history. A single-causality model is used in this work to observe friction form the velocity. It is shown that this model can represent the essential and complex characteristics of the surface friction, including the hysteresis, stick-slip, pre-sliding displacement, frictional lag, and varying breakaway force [28]. Microscopic bristles between contacting

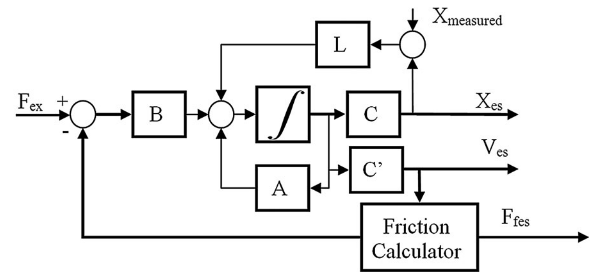

Fig. 15. Block diagram of the observers.

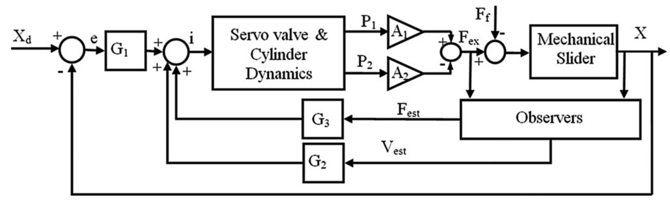

Fig. 16. Block diagram of the control system.

surfaces can be considered to justify the mathematical model. The stick and presliding behavior caused by friction is caused by the deflection of the bristles in the form of small cantilever beams, which will resist the relative motion of the object. After the force passes a specific threshold value (i.e., breakaway force), the bristles deflect sufficiently to allow the object to begin sliding. The mathematical formulations of the friction model are given below [28]–[30]:

$$\frac{dz}{dt} = v - \frac{|v|}{g(v)}z, \quad g(v) = \frac{F_c + (F_s - F_c)e^{-(v/v_0)^2}}{\sigma_0}$$
$$F_f = \sigma_0 z + \sigma_1 \frac{dz}{dt} + \sigma_2 v. \tag{2}$$

Here, v is the relative velocity, F_f is the friction force, F_c is the Coulomb friction, F_s is the stick friction force, v_s is the Stribeck velocity, σ_0 and σ_1 are stiffness and damping coefficients of the bristles, σ_2 represents viscous friction, and z is an internal variable of the model which represents the average bristle deflection.

For the controller, the measured position, observed velocity (outputs for feedback control), and friction (disturbance for feed forward control) are available at each time step. Fig. 16 shows the block diagram of the controller, where the transfer functions G_1, G_2, and G_3 have to be synthesized. The developed tool was then utilized to design these transfer functions. The objective was to make the system to follow a reference model.

The reference model was taken as the following first-order system:

$$G_{\mathrm{Ref}} = \frac{1}{0.02s + 1}. \tag{3}$$

The controller, which was designed in this manner, indicated that the best way to control the system is to compensate for the friction. Specifically, the controller multiplied by the estimated friction G_3 contributes to the current in such a way that the force generated by the current cancels the friction. The Bode plots of

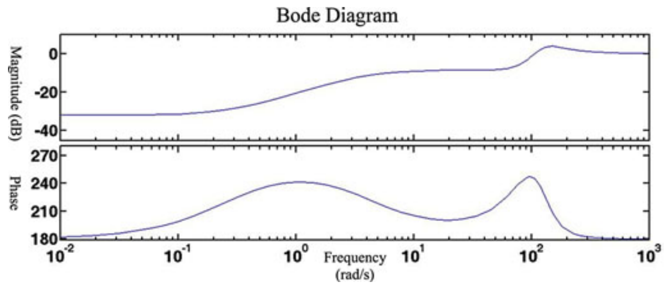

Fig. 17. Bode diagram of the controlled system between friction and output position.

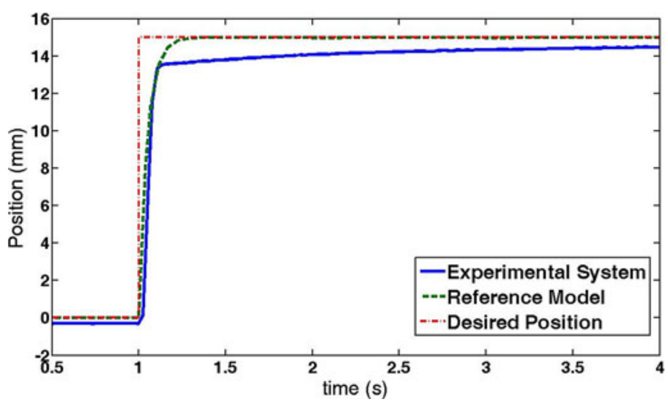

Fig. 18. Performance of the designed controller in comparison with the reference model.

the controlled system between the friction and the output position are depicted in Fig. 17. Since the friction is noticeable in low frequencies, the Bode plot shows that the designed controller is able to successfully attenuate the effect of the friction on the output, thereby compensating for the friction. Therefore, it appears that controller has been synthesized autonomously and intelligently. The performance of the designed controller and the true experimental performance of the system are compared with the desired reference performance, in Fig. 18.

V. Conclusion

Design of a multidomain mechatronic system is a challenging task due to the presence of complex subsystems in different domains and the need to integrate different engineering fields. A bond-graph-based evolutionary system tool was developed in this paper to synthesize mechatronic systems in an optimal manner. A hybrid of GA and GP in a two-loop structure was utilized to mimic the natural evolution of the systems. GP is responsible of the topology realization, while GA is responsible for finding the elite solution of each topology to compete with other elites. The developed tool was used to design analog filters as well as for the controller synthesis for an electrohydraulic manipulator of an industrial fish processing machine—a highly nonlinear mechatronic system. The obtained results were quite encouraging, lending rationale to extend the tool for the development of a more general mechatronic design tool.

References

[1] C. W. de Silva, *Mechatronics – An Integrated Approach*. Boca Raton, FL: CRC Press, Taylor & Francis, 2005.
[2] S. Behbahani and C. W. de Silva, "Mechatronic modeling and design," in *Mechatronic Systems—Devices, Design, Control, Operation, and Monitoring*. Boca Raton, FL: CRC Press, Taylor & Francis, 2007.
[3] S. Behbahani, "Practical and Analytical Studies on Development of Formal Evaluation and Design Methodologies for Mechatronic Systems," Ph.D. thesis, Mech. Eng. Dept, Univ. British Columbia, Vancouver, BC, Canada, 2006.
[4] S. Behbahani and C. W. de Silva, "System-based and concurrent design of a smart mechatronic system using the concept of mechatronic design quotient (MDQ)," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 1, pp. 14–21, Feb. 2008.
[5] S. Behbahani and C. W. de Silva, "Mechatronic design quotient (MDQ) as the basis of a new multicriteria mechatronic design methodology," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 2, pp. 227–232, 2007.
[6] C. W. de Silva, "Sensing and information acquisition for intelligent mechatronic systems," in *Proc. Symp. Inf. Transition*, Hefei, China, Nov. 2003, pp. 9–18.
[7] Q. Li, W. J. Zhang, and L. Chen, "Design for control—A concurrent engineering approach for mechatronic system design," *IEEE Trans. Mechatronics*, vol. 6, no. 2, pp. 161–169, Jun. 2001.
[8] A. C. Pil and H. H. Asada, "Integrated structure/control design of mechatronic systems using a recursive experimental optimization method," *IEEE/ASME Trans. Mechatronics*, vol. 1, no. 3, pp. 191–203, Sep. 1996.
[9] H. M. J. Van Brussel, "Mechatronics —A powerful concurrent engineering framework," *IEEE/ASME Trans. Mechatronics*, vol. 1, no. 2, pp. 127–136, Jun. 1996.
[10] S. F. Alyaqout, P. Y. Papalambros, and A. G. Ulsoy, "Combined robust design and robust control of an electric dc motor," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 3, pp. 574–582, Jun. 2011.
[11] K. Seo, Z. Fan, J. Hu, E. D. Goodman, and R. C. Rosenberg, "Toward a unified and automated design methodology for multi domain dynamic systems using bond graphs and genetic programming," *Mechatronics*, vol. 13, no. 8-9, pp. 851–885, 2003.
[12] J. Hu, E. Goodman, and R. Rosenberg, "Topological search in automated mechatronic system synthesis using bond graphs and genetic programming," in *Proc. 2004 Amer. Control Conf.*, Boston, Massachusetts, pp. 5628–5634.
[13] Z. Fan, K. Seo, J. Hu, E. D Goodman, and R. C Rosenberg, "A novel evolutionary engineering design approach for mixed-domain systems," *Eng. Optimization*, vol. 36, no. 2, pp. 127–147, 2004.
[14] J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane, "Synthesis of topology and sizing of analog electrical circuits by means of genetic programming," *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2, pp. 459–482, 2000.
[15] J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane, *Genetic Programming III, Darwinian Invention and Problem Solving*. San Francisco, CA: Morgan Kaufman, 1999.
[16] J. Wang, Z. Fan, J. P. Terpenny, and E. D. Goodman, "Knowledge interaction with genetic programming in mechatronic systems design using bond graphs," *IEEE Trans. Syst. Man. Cybern. C, Appl. Rev.*, vol. 35, no. 2, pp. 172–182, May 2005.
[17] J. Wang, Z. Fan, J. P. Terpenny, and E. D. Goodman, "Cooperative body–Brain coevolutionary synthesis of mechatronic systems," *Artif. Intell. Eng. Design, Anal. Manuf.*, vol. 22, pp. 219–234, 2008.
[18] S. Behbahani and C. W. de Silva, "Evolutionary Mechatronic Tool," in *Mechatronic Systems—Devices, Design, Control, Operation, and Monitoring*. Boca Raton, FL: CRC Press, Taylor & Francis, 2007.
[19] M. Grossard, C. Rotinat-Libersa, N. Chaillet, and M. Boukallel, "Mechanical and control-oriented design of a monolithic piezoelectric microgripper using a new topological optimization method," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 1, pp. 32–45, Feb. 2009.
[20] Y. Zhang, M. Cong, D. Guo, and D. Wang, "Design optimization of a bidirectional microswimming robot using giant magnetostrictive thin films," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 4, pp. 493–503, Aug. 2009.
[21] Introduction, Genetic Algorithms. [Online]. Available http://cgm.cs.mcgill.ca/∼soss/cs644/projects/marko/introduction.html
[22] J. J. Granda, "The role of bond graph modeling and simulation in mechatronics systems," *Mechatronics*, vol. 12, no. 9–10, pp. 1271–1295, 2002.
[23] D. Karnopp, D. L. Margolis, and R. C. Rosenberg *System Dynamics: Modeling and Simulation of Mechatronic Systems*, New York: Wiley, 2000.

[24] F. O. Karray and C. W. de Silva, *Soft Computing and Intelligent Systems Design*. New York: Addison-Wesley, 2004.
[25] C. W. Tsai, C. L. Lin, and C. H. Huang, "Microbrushless dc motor control design based on real-coded structural genetic algorithm," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 1, pp. 151–159, 2011.
[26] The GP Tutorial. [Online] Available http://www.geneticprogramming.com/Tutorial/
[27] C. W. de Silva and N. Wickramarachchi, "An innovative machine for automated cutting of fish," *IEEE/ASME Trans. Mechatronics*, vol. 2, no. 2, pp. 86–98, Jun. 1997.
[28] C. Canudas de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Trans. Automatic Control*, vol. 40, no. 3, pp. 419–425, 1995.
[29] T. H. Lee, K. K. Tan, and S. Huang, "Adaptive friction compensation with a dynamical friction model," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 1, pp. 133–140, Feb. 2011.
[30] E. J. Berger, "Friction modeling for dynamic system simulation," *Appl. Mech. Rev.*, vol. 55, no. 6, pp. 535–577, 2002.

Saeed Behbahani received the B.Sc. and M.Sc. degrees from the Mechanical Engineering Department, Amir Kabir University of Technology, Tehran, Iran, in 1996 and 1998, respectively, and the Ph.D. degree from the Mechanical Engineering Department, University of British Columbia, Vancouver, BC, Canada, in 2007.

He was a Professor at the Shahid Chamran University of Ahwaz for five years. He is currently an Assistant Professor in the Mechanical Engineering Department, Isfahan University of Technology (IUT), Isfahan, Iran.

Clarence W. de Silva (F'98) received Ph.D. degrees from Massachusetts Institute of Technology, Cambridge, in 1978, and the University of Cambridge, Cambridge, U.K., in 1998, and the Honorary D.Eng. degree from the University of Waterloo, Waterloo, ON, Canada, in 2008.

He has been a Professor of Mechanical Engineering and NSERC-BC Packers Chair in Industrial Automation at the University of British Columbia, Vancouver, BC, Canada, since 1988, where he currently occupies the Tier 1 Canada Research Chair in Mechatronics and Industrial Automation. He has authored 20 books and more than 400 papers, approximately half of which are in journals. His recent books published by Taylor & Francis/CRC are: *Mechatronics—A Foundation Course* (2010); *Modeling and Control of Engineering Systems* (2009); *Sensors and Actuators—Control System Instrumentation* (2007); *VIBRATION—Fundamentals and Practice* (2nd ed., 2007); *Mechatronics—An Integrated Approach* (2005); and by Addison Wesley: *Soft Computing and Intelligent Systems Design—Theory, Tools, and Applications* (with F. Karray, 2004).

Dr. de Silva is a Fellow of the American Society of Mechanical Engineers (ASME), the Canadian Academy of Engineering, and the Royal Society of Canada.