Design Evolution of Engineering Systems Using Bond Graphs and Genetic Programming

by

Buddhika Samarakoon Mudiyanse

B.Sc., University of Moratuwa, 2009

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

The Faculty of Graduate Studies

(Mechanical Engineering)

The University of British Columbia
(Vancouver)

March 2012

©Buddhika Samarakoon Mudiyanse, 2012

Abstract

Engineering design is a complex task, which typically involves multiple physical domains. It can benefit from a modeling tool that can represent different domains in a unified manner. For complex designs, optimization by traditional techniques (such as gradient-based methods) may not be appropriate. Evolutionary methodologies may be used in design optimization of complex engineering systems. This research is based on a framework for evolutionary design system consisting of a machine health monitoring system, model-based evolutionary design optimization system, and a design expert system. The design weaknesses and faults of an existing engineering system are identified using a machine health monitoring system. Then under supervision of the design expert system, an optimal design is evolved using genetic programming. This thesis primarily addresses the modeling and evolutionary aspects and their integration. The thesis develops the integrated system consisting of bond-graph modeling and genetic programming. The performance of the developed system is studied using both experimentation and simulation. The drawbacks of the fitness calculation methodologies that are presented in literature are identified and improved fitness functions are developed in the present work. A methodology to automatically obtain the state-space model of a system represented using bond graphs is also developed. While previous researchers have investigated the integration of bond graphs and genetic programming in design, they have not applied the method in a real engineering system. The present work specifically addresses the application of the developed method for design improvement for an industrial machine. For this purpose a linear bond graph model of the industrial fish processing machine is developed and the parameter values are identified using genetic programming. The design of the actual system is modified according to the evolved bond graph model and the results are validated using the data from the actual engineering

system. The proposed method is applicable particularly to actual systems, first because the initial model can be tested by comparing its simulated results with the corresponding results from the actual system, and second because the design improvements as suggested by the evolutionary design framework may be implemented and tested against the behaviour of the corresponding model.

Preface

This thesis presents research conducted by Buddhika Samarakoon Mudiyansele, under the guidance of Dr. Clarence W. de Silva and Dr. Lalith Gamage. The work presented in part of Chapters 2 and 3 are published in the following conference article. Specifically the sections 2.1.2, 3.1.1 and 3.1.2 contained the methodology and the simulation results presented in this publication.

B. Samarakoon, L. B. Gamage, and C. W. de Silva. Controlled evolution of engineering systems using bond graph and genetic programming. In *Proceedings of the 23rd Canadian Congress of Applied Mechanics*, 2011.

The work presented in this publication was performed by Buddhika Samarakoon Mudiyansele, including designing and implementing the proposed methodologies, performing all simulation experiments. This work was conducted with the guidance and editorial inputs from Dr. Clarence W. de Silva and Dr. Lalith Gamage.

Table of Contents

Abstract	ii
Preface.....	iv
Table of Contents	v
List of Tables	vii
List of Figures	viii
Glossary	xi
Acknowledgments.....	xii
Chapter 1 Introduction	1
1.1. Motivation	1
1.2. Evolutionary Design Framework	3
1.3. Previous Work.....	5
1.3.1. Dynamic System Modeling.....	5
1.3.2. Bond Graph and Genetic Programming.....	6
1.4. Research Objectives	8
1.5. Thesis Outline	9
Chapter 2 Methodology	11
2.1. Bond Graph Modeling and Justification	11
2.1.1. Causality Assignment	13
2.1.2. State-Space Formulation	14
2.2. Genetic Programming	22
2.2.1. Embryo Specification Construction Functions	23
2.2.2. Genetic Operations.....	28
2.2.2. Selection for a Genetic Operations	30
2.3. BG-GP Evolution	32
Chapter 3 Simulation Results.....	35
3.1. An Example in Mechanical Domain	35
3.2. Examples of Electrical Systems	37
3.2.1. Band-stop Filter	37
3.2.2. Band-pass Filter	38

3.2.3	Drawbacks in Fitness Calculation.....	40
3.3	Identification of a System with Desired Eigenvalues	49
Chapter 4	Experimental Validation	52
4.1	Experimental Setup	53
4.1.1	Bond Graph Model	55
4.1.2	Equations of Motion	57
4.1.3	BG Model.....	59
4.1.4	Parameter Identification.....	63
4.2	Design Improvement	65
Chapter 5	Conclusion.....	71
5.1	Summary and Significance of the Contributions	72
5.2	Possible Future Directions	73
Bibliography		75

List of Tables

Table 2.1: Effort and flow in different energy domains.	12
Table 2.2: Bond graph representation of different elements.....	15
Table 2.3: State variables.....	15
Table 2.4: Function set.....	24
Table 2.5: Node and the construction functions for the GP tree in figure 2.8.	25
Table 3.1: Fitness values for systems A and B using the method in [Fan et al., 2004].	41

List of Figures

Figure 1.1: Evolutionary design framework.	4
Figure 1.2: Evolved high-pass filter [Fan et al., 2004].	9
Figure 2.1: Common bond graph model of an electrical circuit and a mechanical system.	12
Figure 2.2: Causality for 1-port element (Power flows from B to A. Effort is the input to A and output to B. Flow is the input to B and output to A).....	14
Figure 2.3: Causalities at a junction.....	14
Figure 2.4: An electrical system and its bond graph.....	16
Figure 2.5: Bond graph with a transformer.....	18
Figure 2.6: A system with linearly dependent states.	20
Figure 2.7: Embryo model.	25
Figure 2.8: Modified system obtained using GP tree in figure 2.9.....	26
Figure 2.9: GP tree created using the construction functions for the embryo in figure 2.7.	27
Figure 2.10: Crossover operation.....	29
Figure 2.11: Mutation operation: (a) selected parent; (b) randomly created GP tree; (c) new offspring.	31
Figure 2.12: Graph of equation 2.18 for different k values.	32
Figure 2.13: Proposed BG-GP flow chart.....	33
Figure 3.1: Embryo for the mass spring damper system.	36
Figure 3.2: Absolute error of the actual and desired frequency response functions as used in fitness calculation.	36
Figure 3.3: Evolved system and the fitness improvement during evolution for the mass spring damper system: (a) evolved system; (b) fitness evolution.....	37
Figure 3.4: Band stop filter evolution, (c) frequency response of the evolved filter.	38

Figure 3.5: Desired frequency response function of a band pass filter with 2nd order butterworth approximation.....	39
Figure 3.6: Band-pass filter evolution.	39
Figure 3.7: Two intermediate solutions during band-pass filter evolution.	40
Figure 3.8: Frequency response for systems A and B in figure 3.7.....	41
Figure 3.9: Weights for frequency response error in different frequency ranges.	42
Figure 3.10: Fitness calculations using a logarithmic scale with different number of samples (n_s) in the range of $1-10^7$ rad/s	44
Figure 3.11: Results for logarithmic scale for the range $1-10^6$ rad/s with 100k samples.	45
Figure 3.12: Evolved band-pass filter and FRF using log scale with 100k samples in frequency range $1-10^7$ rad/s; (a) circuit; (b) frequency response.....	45
Figure 3.13: An evolved system using linear scale with 100 rad/s sampling period: (a) circuit; (b) frequency response.....	47
Figure 3.14: Results using linear scale with different frequency ranges at sampling period 10 rad/s.....	48
Figure 3.15: Two different topologies obtained for evolution of band-pass filter using frequency range $1-9 \times 10^5$ rad/s and sampling interval 10 rad/s: (a) an evolved circuit; (b) another evolved circuit.....	49
Figure 3.16: Identifying a system with specified eigenvalues: (a) circuit bond graph, (b) fitness evolution.	51
Figure 4.1: The iron butcher.	53
Figure 4.2: Conveyor system of the iron butcher.	54
Figure 4.3: Schematic diagram of the linkage.	55
Figure 4.4: Experimental validation procedure.	55
Figure 4.5: Bond graph model of the nonlinear conveyor system.....	56
Figure 4.6: An instance of the table moving to left (f_4 and f_6 are the reaction forces exerted on link AD by CB and ED).	57
Figure 4.7: Bond graph model.	59
Figure 4.8: Unfiltered and filtered acceleration signals.....	64

Figure 4.9: Acceleration profiles of the actual system and the parameter identified model at different frequencies.	66
Figure 4.10: Embryo model.	67
Figure 4.11: Fitness value calculation.	67
Figure 4.12: Modified linkage as prescribed by the proposed BG-GP methodology.....	68
Figure 4.13: Accelerations of the actual system after modification and the evolved BG model.....	69
Figure 4.14: (a) Evolved GP tree representing k_{AD} ; (b) fitness improvement during design evolution.	70

Glossary

Abbreviations

GP	Genetic programming
BG	Bond graph
IIR	Infinite impulse response
TF	Transfer function
GY	Gyrator
FRF	Frequency response function
IB	Iron butcher
VFD	Variable frequency drive
MHMS	Machine health monitoring system

Symbols

e	Effort
f	Flow
S_e	Effort source
C_e	Coefficient matrix
R	Resistor
I	Inductor
C	Capacitor
$\dot{x}$	Time derivative of x
J_g	Gear inertia
ω_m	Motor angular velocity
τ_m	Motor torque
r_g	Gear box ration
τ_g	Gear torque
τ_{k_1}	Torque in the shaft coupling gear box and the disc
r_w	Wheel radius
τ_i	Torque at the input of the gear box
τ_0	Torque at the output of the gear box
J_d	Disc inertia
f_b	Viscous friction force
v_m	Linear velocity of the link
ω_i	Input angular velocity of the gearbox
ω_o	Output angular velocity of the gearbox
J_{AD}	Moment of Inertia of link AD around point A

Acknowledgments

This research and the thesis would not have been possible without the guidance and help of many individuals who contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost, my utmost gratitude is extended to Professor Clarence W. de Silva, my supervisor, for his ample guidance, patience, prompt advices and cooperation. The knowledge which he bequeathed was inspirational. I am deeply indebted to him for funding me during my stay at UBC and providing necessary equipment and laboratory facilities to conduct the experiments and the research presented in this thesis.

I would also like to thank Dr. LalithGamage, Managing Director and the President of Sri Lanka Institute of Information Technology for recruiting me for this research initially. His mentoring during this research and the valuable suggestions for the preparation of this thesis was extremely helpful.

My sincere thanks go to past and present members of the Industrial Automation Laboratory of UBC, including but not limited to Gamini, Tahir, Ramon, Edward, Roland, Shahram and Ibrahim. I am thankful to Dr. SaeedBehbahani, a former PhD student of Prof. de Silva for his discussions and support for my research.

Financial support for this research has come from research grants held by Professor Clarence de Silva, specifically from, Tier 1 Canada Research Chair in Mechatronics and Industrial Automation, Natural Sciences and Engineering Research Council, Canada Foundation for Innovation, and BC Knowledge Development Fund.

Last but not least I would like to thank my parents for the moral support, encouragement, and stimulation proffered throughout my life.

Chapter 1 Introduction

In this introductory chapter, the necessary background information is outlined for the research carried out in the thesis. The objectives, motivation, and rationale of the proposed research are indicated. The literature pertinent to the proposed work is surveyed along with the prior work in our laboratory that set the stage for the present work. What is presented in the remaining chapters of the thesis is summarized as well.

1.1. Motivation

Engineering design starts with an identification of a need and creating new things to improve the human society [Smith, 2000]. In this process, engineers consider different types of constraints such as those related to material, technology and environment [Pahl and Beitz, 1996]. Engineering design is not confined to a particular engineering domain. It may encompass such fields as electrical, mechanical, material and production engineering when it comes to product design. Furthermore, a design engineer must also consider the social and economic aspects during the design process. Given these constraints, engineering design can be considered as a search for a practical solution in a large search space which involves different engineering domains as well as other socioeconomic considerations. The higher the complexity of the problem, the higher the search space for a particular design. In this context, engineering design can be modeled as a search and optimizing problem. As a solution for this problem evolutionary design has been proposed by the engineering design community in the recent past.

Search and optimization algorithms can be divided into three main categories as enumerative, deterministic, and stochastic [Coello et al., 2007]. Enumerative methods consider all possible solutions in the search space and find the optimal solution. Deterministic methods use the domain knowledge such as the gradient field for finding the optima. For example, when using gradient information, the solution will always travel

towards the steepest ascent or descent. Both these methods are inefficient for multi-objective optimization and cannot be used with discrete, non-differentiable, and qualitative variables. Some of these methods are not effective in the presence of local optima as the solution can be trapped in one of them without reaching the global optimum. Stochastic methods employ random initialization and random search steps while allowing inferior solutions within a search, so that they do not trap in local optima. Genetic programming falls into the category of stochastic search and optimization algorithms and they are inspired by biological evolutions. Genetic programming (GP) uses tree structures to represent solutions. Therefore using GP we can represent infinite number of complex solutions and non-analytic solutions [Wang et al., 2005, Fan et al., 2004]. Instead of considering one solution at a time, GP considers a population of feasible solutions together. Possible solutions in GP are represented using graphs (trees) which can be used to explore a large search space. Engineering designs represented by graphs can be evolved using GP to explore the design space and to find the optimal candidates.

In the work presented in this thesis, an evolutionary design approach is used for automatically generating an optimal proposal for design improvement of an industrial machine. Conventional methodologies for engineering design usually use a sequential approach. In them, the design of a particular segment of the system that belongs to a domain is carried out separately and the overall design of the system is done in a sequential manner. These design procedures require precise input-output specifications for each design subset and work within a limited design space for each subset. With difficulties in mathematical analysis and simultaneously achieving multiple objectives, this process becomes tedious in modern engineering design. Also due to dynamic interactions or coupling between domains or design subsets, the overall design generated by a sequential procedure will be non-optimal in general. Since most present day engineering systems are multi-domain systems, their designing should use an integrated and concurrent approach encompassing all relevant domains and subsystems. The design optimization should consider all significant dynamic interactions within the subsystems. This understanding, as expressed by others, is the basis of the preliminary work carried out in the research leading to this thesis [Samarakoon et al., 2011]. The overall system

framework was proposed by de Silva in 2008 [de Silva, 2008; de Silva 2010] which consistsof a machine health monitoring system, an automated modeling tool, and an evolutionary optimization procedure under the supervision of a design expert system. The proposed method in this work is based on that original framework, and formulating the engineering design problem as a search and optimizing problem using evolutionary algorithms. This approach allows using an infinite search domain and it enables the systematic use of monitored information and expert knowledge [Gamage and de Silva, 2010]. Often in engineering design we learn from the failures of previous systems, and from the operating information of the current systems [Bar-Yam, 2003]. The proposed methodology in this thesis allows the design engineer to use this information in his engineering design process. This thesis presents a methodology using genetic programming with bond graph modeling for design evolution of an industrial system.

1.2. Evolutionary Design Framework

This research is based on Evolutionary Design System (figure 1.1) proposed by [de Silva, 2008] and further enhanced in [Gamage and de Silva, 2010; de Silva 2010]. In this framework, there is an industrial machine connected to a health monitoring system which can identify faults and detect malfunctions [Raman and de Silva, 2009; Razavi and de Silva, 2010]. These faults or performance degradation can be due to wear and tear or design weaknesses of the industrial machine. In this thesis the focus is the “faulty design” problem where design weaknesses are identified and remedies are proposed. From sensory data, the machine health monitoring system is able to not only detect performance degradation but also its cause (specific components associated with the degradation), which is a “diagnosis” exercise. In the original system, only component faults or malfunctions are diagnosed. In the extended system, which is incorporated in the present work, design weaknesses (design faults) and associated system components or segments are also diagnosed with the assistance of a design expert system. In this manner, faulty components or subsystems are isolated for design improvement. This subsystem is automatically redesigned using evolutionary design methodologies of the

proposed system. During this phase, the design of a subsystem is modeled using a multi-domain modeling methodology and the design is evolved using genetic programming. A solution should not be allowed to evolve freely as with conventional evolutionary optimization schemes [Karray and de Silva, 2004] because all the evolve design may not be feasible for actual implementation. An expert system has the capability of providing inputs to the evolutionary subsystem and these inputs can be incorporated to guide the design evolution. This is the concept of “supervised” design evolution [de Silva, 2010].

Figure 1.1: Evolutionary design framework.

The advantages of this evolutionary design framework are that it allows the systematic use of monitored information for design improvements and the use of knowledge of previous designs through an expert system. The use of genetic programming for design evolution avoids complex mathematical analysis and the need for precise input/output specifications while making multiple objectives achievable. Component costs and system complexity are some of the concerns a design engineer may have to address in designing an engineering system. The research presented in this thesis mainly focuses on the evolutionary design system which uses genetic programming and

bond graph modeling. The evolutionary process is controlled to evolve designs that are practically implementable using the available resources.

1.3. Previous Work

1.3.1. Dynamic System Modeling

A dynamic system is a system in which its states are a function of time and have non-negligible rates of change [de Silva, 2005]. Modeling plays an important role in designing a dynamic system. For the control and simulation purposes it is better to have a detailed model of the system, while it should be simple enough to analyze and implement. Modeling methodologies should provide space to make improvements as one progress through a design phase of a particular system. In addition to these features most important aspect of a modeling technique is to facilitate for modeling multi-domain systems [de Silva, 2009].

Even though structurally non-specific analytical models are good for mathematical analysis they may not provide sufficient information about the interaction between components. If the designer needs to work on some specific subsystems, structurally non-specific analytical models such as transfer functions are not helpful. Since a transfer function relates only input and output, the internal system structure and behaviour cannot be obtained through a transfer function. The other modeling methodologies like block diagrams and signal flow graphs [Mason, 1953] are not helpful in multi-domain system representation as well as giving a unique model for a particular system [de Silva, 2005; Gawthrop and Bevan, 2007]. For example, the Direct Form I and II of an IIR filter are two realizations of the same function. Another drawback is that it is difficult to make a physical correspondence between the model and the actual system when using structurally non-specific analytical models.

Linear graphs and Bond graphs [de Silva, 2005] are good choices for modeling multi-domain systems and graphical representation of them. Linear graphs were introduced by [Firestone, 1933] and they use connected graph to model the dynamic

systems. The main variables represented in a linear graph are through variables and across variables. As the name implies, a through variable is a quantity that goes unchanged through an element, like current and force. On the other hand across variable is a quantity measure across an element such as voltage and velocity. Bond graphs were introduced in early 1960s by Henry Paynter of Massachusetts Institute of Technology [Paynter, 1960] which uses effort and flow variables of different energy domains to achieve multi-domain modeling. Bond graphs have been used for modeling engineering systems for the last half-century [de Silva, 2006; Karnopp et al., 2006] and widely accepted for representing mechatronic systems. In comparison to linear graph modeling, bond graph modeling is a well established technique within the research community which is supported by a range of software tools and documentation. Also, BG modeling provides convenient ways to choose state variables to build mathematical models and to convert them into signal flow graphs [Borutzky, 2010]. The causal properties of a bond graph is useful for finding independent states of a system and hence to simplify the model. Bond graphs were further extended for modeling linear distributed systems using normal modes by [Margolis, 1985]. Later, researches have worked extensively on bond graphs and methods in determining controllability and observability [Sueur and Dauphin-Tanguy, 1991]. A notable advantage of bond graphs is that there are well developed software packages that can simulate bond graphs. Borutuzky [Borutzky, 2002] explains how the bond graph modeling closely resembles the concepts of object oriented modeling. There are free software packages that can be used for modeling and simulating dynamic systems using bond graphs [Broenink, 1999]. Recently, bond graphs have been used for modeling complex systems like backhoe machines [Krishnaswamy and Li, 2006] and robot manipulators [McPhee, 2005].

1.3.2. Bond Graph and Genetic Programming

Evolutionary processes for artificial systems were initially proposed by John Holland in his book “Adaption in Natural and Artificial Systems” [Holland, 1975]. In the early days,

evolutionary algorithms were used in the area of artificial intelligence for evolving finite state machines [Back et al., 1997]. With the advancement of computers, genetic programming was applied for automatic program creation by representing individuals of a population as a tree comprising of primitive functions [Koza, 1995]. Later, evolutionary computation was used in optimization problems in the domains of digital filter design, symbolic regression, control systems and robotics [Coello, 1999; Watanabe and Izumi, 1999]. In the late 1980s [Kristinsson and Dumont, 1988] used genetic algorithms for identifying poles, zeros and gain of a controller. The use of evolutionary techniques is advantageous in system identification since it generally involves a high number of dimensions and multi-modality. These methods have now been extended even for non-linear system identification. Power of genetic programming in identifying new systems has been explained in the work of [Koza et al., 2004]. Their work also explains how the patented inventions were reinvented using genetic programming-based evolutionary design.

In the mid to the late 1990s [Grimbleby, 1995, Koza et al., 1997] proposed genetic programming for synthesizing electrical circuits. In the work of Koza et al., they did not use any modeling technique for the circuits, and the solutions were evaluated using SPICE simulations [Quarles et al., 1994]. The individuals of a population represent a circuit construction tree which will be evolved using genetic programming. This was a milestone in evolutionary design. Subsequently researchers worked on using evolutionary techniques for system design, which maybe termed evolutionary design. Tay and colleagues [Tay et al., 1998] used BG and genetic algorithms for the design modification of an air pump. It was a multi-domain representation as it modeled the motor and the pump lever assembly together. In their work, the construction of a bond graph was represented in two-dimensional binary patterns. [Fan et al., 2004] introduced genetic programming for multi-domain engineering design. This work has also been used in the field of Micro-electro-mechanical Systems (MEMS) design. However, their work has not addressed the problem of complexity of the evolved system. For example, the high pass filter shown in Figure 1.2 has an order greater than 10, even without considering the sharp transition from stop band to pass band. The BG-GP methodology was also widely

employed in designing vibration absorbers. Using the BG-GP methodology [Hu et al., 2008] have obtained results that are comparable with those from conventional design methodologies, for vibration absorbers. Nonetheless, much of the BG-GP work presented in the literature have not been tested with real world industrial machines. The work of [Behbahani and de Silva, 2008] proposed a multi-criteria index for system evaluation and applied it for selecting a transmission system for a fish cutting machine. The work presented in this thesis relies on the framework presented in Section 1.2 while using bond graphs and genetic programming for iterative design improvements in an industrial machine.

1.4. Research Objectives

Given this background, the objectives of the research presented in this thesis can be summarized as follows.

1. Develop a tool for automated and controlled evolution of an existing mechatronics system using bond graph (BG) modeling and genetic programming (GP).
2. Validate the proposed methodology for an industrial machine.

The BG-GP evolution is used to identify both topology and parameters of an improved system while considering the complexity and the feasibility of implementation. The test bed for this methodology will be the fish cutting machine (Iron Butcher) developed at the Industrial Automation Laboratory (IAL) of the University of British Columbia.

Figure 1.2: Evolved high-pass filter [Fan et al., 2004].

1.5. Thesis Outline

This thesis contains five chapters entitled Introduction, Methodology, Simulation Results, Experimental Validation and Conclusion. **Chapter 1** introduced the subject of the research, provided the objectives and rationale, and surveyed the pertinent literature. An outline of the next four chapters is given below.

Chapter 2 presents the approach of design evolution. First, bond graph modeling is introduced and then a simulation methodology for state space models that are modeled through bond graphs is given. Subsequently a genetic programming-based design evolution is presented.

Chapter 3 contains the simulation results obtained from the proposed BG-GP based evolutionary design methodology. Identification of a mass-spring-dampers system and of electrical filters is presented in this chapter. The drawbacks of the fitness calculation methods proposed in previous work are identified, and improved fitness functions are proposed in order to obtain results in a more reliable manner.

Chapter 4 presents the experimental validation of the proposed methodology. A bond graph model of the Iron Butcher has been developed and the parameter values are identified using genetic programming. Finally, a linkage of the machine is modified to obtain the required acceleration, as proposed by the machine health monitoring system. It is validated using the actual acceleration of the newly designed linkage.

Chapter 5 concludes the work presented in this thesis while mentioning significant contribution and possible future work related to this research.

Chapter 2 Methodology

The goal of the methodology that is proposed in this thesis is to evolve an existing system into a new and optimized design that satisfies a set of performance requirements. This evolutionary design optimization process is able to generate optimal topologies and components so as to satisfy the design requirements. In addition, the evolutionary algorithm should be able to incorporate such design constraints as complexity, components costs, component availability and implementation feasibility. There are two desirable features in the developed evolutionary design optimization procedure. First, there should be a methodology to represent the engineering system. Second, an evolutionary algorithm has to be used in order to evolve an optimal design of the engineering system. As described in Chapter 1, bond graph modelling is used for representing these engineering systems and genetic programming has been used for evolutionary optimization. These two features are explained in following sections.

2.1. Bond Graph Modeling and Justification

As described in the previous chapter, bond graph modeling is a multi-domain, unified modeling technique which is useful in modeling and simulation of a mechatronic system. Numerous researchers have worked on bond graphs and there exist well developed methodologies for representation and computer simulation of multi-domain systems using bond graphs. The unified and standardized nature of bond graphs is shown in Figure 2.1, in which two systems in electrical and mechanical domains have been modeled by a common bond graph. There are two main variables in a bond graph, which are called effort and flow. The bond graph represents the interaction of these two variables between interconnected elements in the system. The effort and flow variables in different energy domains are given in Table 2.1. Regardless of the domain, the product of these two variables equals to power.

Figure 2.1: Common bond graph model of an electrical circuit and a mechanical system.

Table 2.1: Effort and flow in different energy domains.

Domain	Effort	Flow
Mechanical	Force	Velocity
Electrical	Voltage	Current
Fluid	Pressure	Flow Rate

In a bond graph, there are two main components, which are called junctions and bonds. The bonds are the line segments with half arrows which indicate the direction of power flow. For a source this power flow will always be out of the source and towards the system. Therefore, the half arrow will be directed outwards from the source element. Since a passive element will always consume energy, its half arrow will be directed towards that element from the system. A junction in a bond graph indicates which variable (effort or flow) is common to the components attached to that junction. The net power flow into any of these junctions is equal to zero. In a 1 junction the flow variable of

the components attached to that junction is common while in a 0 junction the effort will be common. In Figure 2.1 velocities of mass m_2 , spring k_2 and viscous damper b_2 are the same. These three components are analogous to inductor L_2 , capacitor C_2 and resistor R_2 in the circuit, which have a common current through them. They are attached to the last 1 junction of the bond graph. Similarly one can find the analogy between other elements and their bond graphs.

2.1.1. Causality Assignment

As mentioned in Chapter 1 causality is an important property in a bond graph. It assists the modeler in formulating a state-space model for the system and in identifying independent states of the system. Causality specifies the input and the output of a particular element. In the bond graph context this input or the output will be the flow variable or the effort variable associated with that element. From effort and flow, if one variable is the input at a particular side of the bond, that same variable will be the output at the other side of the bond.

There are two forms of causality called integral causality and differential causality. In integral causality, the integral form of the constitutive equation of the element is used. For example in a capacitive element the constitutive equation in integral form is as given in Equation 2.1, where e and f are the effort and flow variables. Often integral causality is preferred in bond graph modeling since it assists in obtaining state-space models and the corresponding ordinary differential equations [de Silva, 2005; Gawthrop and Bevan, 2007]. Figure 2.2 shows two elements connected in a bond with integral causality.

$$e = \frac{1}{c} \int f dt \quad (2.1)$$

The bond graph representations of different elements used in the present work, along with their causalities are given in Table 2.2.

Figure 2.2: Causality for 1-port element (Power flows from B to A. Effort is the input to A and output to B. Flow is the input to B and output to A).

For a common flow junction, since all the flow variables associated with the junction are the same, only one variable can be the input. Similarly in a common effort junction, since all the efforts are the same only one effort variable can be the input. With these rules, possible causalities for junctions are shown in Figure 2.3. Causality assignment for this work has been adapted from [de Silva, 2005; Behbahani and de Silva, 2007].

Figure 2.3: Causalities at a junction.

2.1.2 State-Space Formulation

In an evolutionary process, which will be presented in the next section, measuring how well an individual can perform the required task is an important part. In order to measure this in the proposed methodology the state space model of a bond graph is obtained and it is simulated in order to obtain the responses in the time domain or the frequency domain. In the rest of this section the procedure for obtaining the state-space model from a bond graph is explained.

In order to extract the state-space model, first the independent energy storage elements have to be identified. In the electrical domain the energy storage elements will be capacitors and inductors and in the mechanical domain they are springs and masses. The state variables for these four elements are given in Table 2.3.

Table 2.2: Bond graph representation of different elements.

Mechanical	Electrical	BG Representation
Force Source	Voltage Source	$S_e \longrightarrow$
Velocity Source	Current Source	$S_f \longleftarrow$
Spring	Capacitor	$\longleftarrow C \longrightarrow$
Inertia	Inductor	$\longrightarrow I \longleftarrow$
Damper	Resistor	$\longleftarrow R \longrightarrow$ $\longrightarrow R \longleftarrow$
Mechanical Transformer	Electrical Transformer	$\longrightarrow \overset{n}{\text{TF}} \longleftarrow$ $\longleftarrow \overset{n}{\text{TF}} \longrightarrow$
Mechanical Gyrator	Electrical Gyrator	$\longleftarrow \overset{n}{\text{GY}} \longrightarrow$ $\longrightarrow \overset{n}{\text{GY}} \longleftarrow$

Table 2.3: State variables.

Capacitor	Voltage	Spring	Force
Inductor	Current	Mass	Velocity

If any of the energy storage elements are attached to a source such that their states are defined by the input, they cannot be selected as independent energy storage elements. For example, connecting a voltage source in parallel with a capacitor (or attaching a force

source to a spring in series) or connecting a current source in series with an inductor (or attaching a velocity source to a mass) will lead to dependent states. After choosing the state variables, the auxiliary variables are also selected, which are required to formulate the state space shell. For each junction in the bond graph a separate variable will be assigned if it is not a state variable. For the junctions, its common variable is taken as an auxiliary variable if it is not a state. For energy storage elements, the variable other than its state will be chosen such that the total auxiliary variables can be minimized. For the system in Figure 2.4a an explicit variable for the voltage across inductor L has not been given since it is equal to v_L which is the common variable in that junction. Similarly, for the current through the capacitor C has not been assigned a separate variable since it is the same as i_2 . The vector $\mathbf{X}$ in 2.2 contains the variables selected in this manner for the system shown in Figure 2.4a. S_e is the input, which is an effort source and v_L , i_L , and i_2 are auxiliary variables. The state variables in this system are v_c and i_L .

$$\mathbf{X} = \begin{bmatrix} \dot{v}_c & \dot{i}_L & v_c & i_L & S_e & v_L & i_L & i_2 \end{bmatrix} \quad (2.2)$$

Figure 2.4: An electrical system and its bond graph.

After selecting these variables, the algorithm will visit all the junctions in the bond graph in order to develop the constitutive and continuity equations. The continuity equation for a common flow junction will be: the sum of all the efforts equal to zero, and for a common effort junction: sum of all the flow will be equal to zero. These equations for the two junctions in Figure 2.3 are,

$$\begin{aligned}e_1 + e_2 + e_3 &= 0 \\f_1 + f_2 + f_3 &= 0 .\end{aligned}$$

These represent the Kirchhoff's current and voltage laws in the electrical domain or the geometric compatibility and dynamic equilibrium of forces in the mechanical domain. These equations for the circuit in Figure 2.4a can be written as follows:

- For the first common flow junction, the continuity equation is:

$$S_e = i_1 R_1 + v_1 \quad (2.3)$$

(Effort variable across the energy dissipating element is written directly by multiplying with the corresponding flow, without using a separate variable)

There will be no constitutive equations for this junction as there are no energy storage elements attached to this junction.

- For the second common effort junction, the continuity equation is:

$$i_1 = i_L + i_2 \quad (2.4)$$

Constitutive equation is:

$$v_1 = L \dot{i}_L \quad (2.5)$$

- For the last common flow junction, the continuity equation is:

$$v_1 = i_2 R_2 + v_c \quad (2.6)$$

Constitutive equation is:

$$i_2 = C \dot{v}_c \quad (2.7)$$

When there are transformers and gyrators their constitutive equations have to be considered when writing the continuity equations. As an example, for the configuration shown in Figure 2.5 the continuity equations should be written as:

$$i_1 = i_L + ni_2 \quad (2.8)$$

$$v_1 = \frac{1}{n}(v_c + i_2 R_2) \quad (2.9)$$

After completing these equations, it is observed that the Kirchhoff's current and voltage laws for the circuit in Figure 2.4a have been automatically taken into account. In order to eliminate the auxiliary variables, equations 2.3-2.7 are formed into a system of linear equation as in 2.10. This matrix will be referred to as the coefficient matrix (C_e) in the sequel.

Figure 2.5: Bond graph with a transformer.

$$\begin{bmatrix} 0 & 0 & -\frac{1}{C} & 1 & 0 & 0 & 0 & 0 \\ -\frac{1}{L} & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & -R_1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & -R_2 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ i_1 \\ i_2 \\ v_c \\ i_L \\ v_c \\ i_L \\ S_e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.10)$$

The auxiliary variables in $\mathbf{X}$ can be eliminated by taking the reduced row echelon form of $\mathbf{C}_e$. For clarity, letting $R_I = 1$, $L = 5$, $C = 4$ and $R_2 = 1$, the reduced form of equation (2.10) can be rewritten as in equation (2.11). This separates the auxiliary variables and gives them as linear differential equations of state variables and inputs. In order to achieve this, there should be $n_s + n_a$ number of equations, where n_s is the number of states and n_a is the number of auxiliary variables. The last n_s equations of the linear system will give the state equations. That are:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0.3333 & -0.6667 & -0.3333 \\ 0 & 1 & 0 & 0 & 0 & -0.3333 & 0.6667 & -0.6667 \\ 0 & 0 & 1 & 0 & 0 & 0.3333 & 0.3333 & -0.3333 \\ 0 & 0 & 0 & 1 & 0 & 0.0833 & 0.0833 & -0.0833 \\ 0 & 0 & 0 & 0 & 1 & -0.0667 & 0.1333 & -0.1333 \end{bmatrix} \begin{bmatrix} v_1 \\ i_1 \\ i_2 \\ v_c \\ i_L \\ v_c \\ i_L \\ S_e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.11)$$

$$\dot{v}_c + 0.0833v_c + 0.0833i_L - 0.0833S_e = 0 \quad (2.12a)$$

$$\dot{i}_L - 0.0667v_c + 0.1333i_L - 0.1333S_e = 0. \quad (2.12b)$$

Hence, the matrices $\mathbf{A}$ and $\mathbf{B}$ of the state-space model are:

$$\mathbf{A} = \begin{bmatrix} -0.0833 & -0.0833 \\ 0.0667 & -0.1333 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0.0833 \\ 0.1333 \end{bmatrix}$$

The state vector is $[v_c i_L]^T$.

The output equations can also be obtained using the reduced C_e matrix. If the output is v_L , the output equation corresponds to the 1st row of the set of equations as given by:

$$v_1 + 0.3333v_c - 0.6667i_L - 0.3333S_e = 0. \quad (2.13)$$

Then the C and D matrices will be:

$$C = [-0.3333 \ 0.6667]; \quad D = [0.3333]$$

If the output variable is not in the vector X it can be indirectly obtained by multiplying with some other component values. If the voltage across R_2 is required, multiplying the equation corresponds to i_2 by R_2 will give that quantity.

2.1.3 Presence of Linearly Dependent States

As mentioned in the previous section, when selecting state variables only the states that depend on the input have been omitted. But there is another scenario in which the states are linearly dependent. A particular scenario is shown in Figure 2.6. Here the states of C_1 and C_2 are linearly dependent through the transformer coefficient n . When there is a transformer, additional variables have to be introduced in order to add the constitutive relationships of the transformer (i_2 in this case). The reduced form of the equation set as obtained using the methodology explained in previous section would give the set of equations in 2.14. The parameter values used are, $R_1 = 5; C_1 = 6; R_2 = 4$ and $C_2 = 2$.

Figure 2.6: A system with linearly dependent states.

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & -0.2 \\
0 & 1 & 0 & 0 & 0 & -4 & 0 & -0.5 & 0 \\
0 & 0 & 1 & 0 & 0 & 4 & 0 & 0.6 & -0.2 \\
0 & 0 & 0 & 1 & 0 & -2 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0.1667 & 0 & 0.1 & -0.0333 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -0.5 & 0
\end{bmatrix}
\begin{bmatrix}
i_1 \\
i_2 \\
i_{c1} \\
i_{c2} \\
\dot{v}_{c1} \\
\dot{v}_{c2} \\
v_{c1} \\
v_{c2} \\
S_e
\end{bmatrix}
=
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix} \quad (2.14)$$

These sets of equations are clearly different from the reduced set of equations in 2.11. The state equations depend on auxiliary variables and furthermore, the equation for $\dot{v}_{c1}$ contains $\dot{v}_{c2}$. These problems arise due to the fact that the dependent state has been chosen as an independent state. In order to remove this dependency, the time derivative of the last equation is added to this system of equations. This type of dependencies will be indicated in the coefficient matrix C_e by having a row which has only non-zero elements in the columns corresponding to state variables. In the present example it is the 6th row. Hence the time derivative of this equation is

$$\dot{v}_{c1} - 0.5 \dot{v}_{c2} = 0. \quad (2.15)$$

This will be added to the system of equations which will give a total of 7 equations. Now the dependent state (v_{c2}) and its derivative ($\dot{v}_{c2}$) become auxiliary variables. Therefore, there are 6 auxiliary variables and only one state variable which is v_{c1} . These numbers agree with the requirement in the previous section which dictates that the total number of equations has to be $n_s + n_a$. The new set of equations is given in 2.17. The variable positions were changed such that the last row would give the only remaining state equation, as given by:

$$\dot{v}_{c1} = -0.0857 v_{c1} + 0.0143 S_e \quad (2.16)$$

$$\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0.2 & -0.2 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & -0.3142 & -0.1142 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0.5143 & -0.08571 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0.3429 & -0.05714 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0.1714 & -0.0286 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & -2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0.0857 & -0.0143
\end{bmatrix}
\begin{bmatrix}
i_1 \\
i_2 \\
i_{c1} \\
i_{c2} \\
\dot{v}_{c1} \\
\dot{v}_{c2} \\
\dot{v}_{c1} \\
\dot{v}_{c1} \\
S_e
\end{bmatrix}
=
\begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix} \quad (2.17)$$

2.2. Genetic Programming

Genetic programming considers a set of solutions which is called a population at any given time [Karray and de Silva, 2005]. To employ genetic programming for a search and optimization problem there should be an efficient way to represent the solutions in a tree-like structure [Koza, 1992]. Three main functions of an evolutionary process are solution representation, fitness evaluation and generation of new populations. In the work presented in this thesis the construction of a bond graph is represented as a tree, which is the solution representation. Then there should be a measurement on how well each individual is able to satisfy the design requirements, which is called the fitness. Definition of the fitness depends on the problem and it plays an important role in an evolutionary process since it guides us ultimately to our goal. Finally the genetic operations are used to evolve a new population from the current population until the required solution is obtained. The very first population of a genetic program is obtained by randomly generating a set of GP trees. This random growth of the initial population is an important feature of genetic programming. Essentially this will consider random samples from the search space for possible solutions. For large search spaces where sampling the entire search space is not feasible, this process will help in avoiding any local optima.

2.2.1. Embryo Specification Construction Functions

To start an evolutionary process, first an embryo model has to be identified. From a system point of view the embryo contains the information about the input and output and the locations that are modified during an evolutionary process. Figure 2.7 shows an embryo model of this type. In this embryo model the 1 junction is a modifiable site which corresponds to a part in the actual circuit shown within the dashed rectangle. The modification that can be done to an embryo in this work are listed below:

1. Inserting an element to a modifiable junction
2. Inserting a junction to a modifiable bond
3. Changing the parameter values of the elements.

These changes themselves will add more modifiable sites to an embryo. For example, when an element is inserted to a junction, it will create a modifiable bond. Also the element's parameter value has to be computed. It can be further modified during evolution.

Construction Functions:

The modifications that are done to modifiable sites are called construction functions. These construction functions are formed into a tree which will eventually be represented as a tree structure. This is the individual representation within the genetic program. The set of construction functions used in this work along with their identification numbers are given in Table 2.4. At the start of an evolution the new individuals are created by randomly using these construction functions. As an example, there is an equal probability in inserting a common flow junction or a common effort junction to a modifiable bond. Similarly, for parameter value computation, arithmetic operations are selected depending on a random number generated by the program. For the parameter value computation, there should also be numerical values as the operands. Even though there can be some pre-defined scaling factors for these values, there should be a random behavior in generating the numerical values.

The GP tree in Figure 2.8 and Table 2.5 shows the construction functions that can be applied to the embryo in Figure 2.7. Node 3 in this GP tree indicates an addition of a resistor to the modifiable junction, which creates another two modifiable sites. One modifiable site is the parameter values of the resistor (node starting from 4). The other modifiable site is the bond which connects resistor and the junction. In this case a common flow junction has been inserted to this modifiable bond (node starting from 19).

Table 2.4: 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

Figure 2.7: Embryo model.

Table 2.5: Node and the construction functions for the GP tree in figure 2.8.

Node	ID
1	0
2	33
3	21
4	12
5	12
6	11
7	11
8	13
9	16

Node	ID
10	11
11	13
12	11
13	13
14	11
15	11
16	13
17	11
18	11

Node	ID
19	31
20	23
21	13
22	11
23	12
24	11
25	16
26	11
27	16

Node	ID
28	14
29	11
30	11
31	11
32	30
33	20
34	20
35	20

Figure 2.8: Modified system obtained using GP tree in figure 2.9.

Bond Graph Representation:

In this thesis a bond graph is represented using a $j_n \times 6$ junction array and $ab_n \times 4$ bond array where, j_n and b_n are the number of junctions and number of bonds connecting junctions, respectively. The 6 columns of the junction array correspond to resistance, inductance, capacitance, effort source, flow source and junction type, respectively. In the bond array, the first column gives the transformer or the gyrator coefficients while the next two columns give the row numbers of the junction array which correspond to two junctions that are connected through that particular bond. The last column indicates whether it is a transformer (1), gyrator (2) or an ordinary bond (0). With this notation the junction and bond arrays for the system in 2.4a can be written as:

$$\begin{bmatrix} R_1 & 0 & 0 & 1 & 0 & 1 \\ 0 & L & 0 & 0 & 0 & 0 \\ R_2 & 0 & C & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 2 & 0 \\ 1 & 2 & 3 & 0 \end{bmatrix}$$

Two separate vectors have to be specified in order to input the modifiable site. First vector contains the type, whether it is a modifiable arithmetic site, bond or a junction. The second vector contains the location in the junction array or bond array. If the

parameter value R_I has to be modified, then the modifiable site will be (1,1), and the type will be arithmetic. If the common effort junction has to be evolved, the modifiable site will be 2 and the type will be junction.

Figure 2.9: GP tree created using the construction functions for the embryo in figure 2.7.

2.2.2 Genetic Operations

A main feature of any evolutionary optimizing method is that it considers a set of solutions in parallel. As it progresses it increases the adaptability of its individuals. In order to achieve this, there should be topological and parametric changes to the individuals, which are achieved by evolving the systems represented by the bond graphs. These changes are done through genetic operations. A new population of offspring is obtained by applying genetic operations to the parent population [Koza, 1992]. Genetic operations are the basic steps of an evolutionary process where the Darwinian natural selection is applied. The selection criterion used in this research is explained later in this section. The three genetic operations used in this work are [Karray and de Silva, 2004]

1. Repetition
2. Crossover
3. Mutation.

Repetition and mutation are asexual operations where only one parent chromosome is required to produce an offspring. The crossover operation requires two parents and it produces two offspring. These genetic operations have to be modified accordingly in order to apply for bond graphs-based design evolutions.

Repetition:

Repetition copies the chromosomes in the parent population to the child population without any alteration. The parent individual is selected according to the selection criterion, and some percentage (p_r) of the child population is obtained by repetition.

Crossover:

Since the repetition directly copies parents in their entirety to the child population, it may lead to premature convergence of the genetic program. Crossover and mutation add diversity to the evolution [Koza, 1992]. Crossover is analogous to sexual reproduction in biological species. In the crossover operation, two branches of the GP tree of two different parents are chosen and they are interchanged to produce two new offspring as

shown in Figure 2.10. This operation is useful for having a better offspring with good features from the two parents. In this research the crossover operation is applied depending on the type of the branch.

(a) Selected chromosomes from the parent population.

(b) New individuals.

Figure 2.10: Crossover operation.

For a bond graph model represented by a GP tree such as the one shown in Figure 2.9 it is meaningless to do a crossover operation without considering the types of the branches. As an example, replacing a branch corresponds to a common flow junction with a branch representing an arithmetic value would not create a meaningful and realistic physical system. Therefore, each time a new population is created, the crossover operation is applied separately among the type of branches that match with each other.

Accordingly, the three types of operations used are element crossover, junction crossover and arithmetic crossover. Furthermore, the arithmetic crossovers are done only with matching types of elements in order to increase the efficiency of the genetic program. For example, in a high frequency filter design problem, most of the capacitors and the inductors may have very low parameter values on the order of approximately 10^{-3} - 10^{-6} F/H, while the resistors might have values in the ranges beyond $1k\Omega$. In this type of a problem, replacing a capacitor value with a resistor would not help the evolution. Hence the arithmetic crossovers are also applied only among matching type of elements.

Mutation:

Mutation introduces a variation to the evolution by adding randomly created branches to the GP tree. This is useful in achieving the global optimum since this operation will introduce new solutions that have not been considered previously. Only one parent is involved in mutation. A node in the GP tree is randomly selected from a uniform distribution and, similar to the crossover, the random generation of the new branch is done depending on the type of node that has been selected. Out of the total offspring set that can be created, different percentages are assigned for different types of branches, so that all the possible cases get a fair chance in mutating. The mutation operation is illustrated in Figure 2.11.

2.2.2. Selection for a Genetic Operations

The selection of an individual for a genetic operation has to be done by applying the Darwinian principles of natural selection. In the present work the candidates in a solution are ranked according to their fitness and the selection for the genetic operations is based on this rank. The main objective of a selection mechanism is to give a higher probability for the individuals having better fitness values. In this research the selection of individuals is based on the equation:

$$S = p_n(1 - r^k) \quad (2.18)$$

where r is a random number uniformly distributed within 0 and 1; $p_n > 0$ is the population size; S is the rank of the individual to be selected for a genetic operation; and $k < 1$ is some number which depends on how much weight should be given to better individuals when selecting for genetic operations. Figure 2.12 shows the effect of k on Equation 2.18 after normalizing ($p_n = 1$). The smaller the value of k , higher is the probability of selecting a better individual for the operations. In this work the population size is kept at a constant value throughout the evolutions.

Figure 2.11: Mutation operation: (a) selected parent; (b) randomly created GP tree; (c) new offspring.

Figure 2.12: Graph of equation 2.18 for different k values.

2.3 BG-GP Evolution

The complete process of the proposed evolutionary design methodology based on bond graph modeling and genetic programming is depicted as a flow chart in Figure 2.13.

First, an embryo model has to be identified to start the evolutionary process. This specifies the inputs and outputs and the modifiable sites of the system. The first population of the evolution will be obtained by randomly growing construction functions, as described in the previous section. These construction functions are formed into tree-like structures as the individual representation within the genetic program.

Next, the GP trees are converted back to bond graphs for simulation purposes. The simulations are done by obtaining the state space models from the bond graphs. Fitness values of the individuals are calculated by simulating these systems in time domain or frequency domain. After simulation, fitness values will be assigned to all the

individuals in a population. A fitness value depends on how well a particular system can perform the required task. These individuals are then ranked according to their fitness.

Figure 2.13: Proposed BG-GP flow chart.

There are two possibilities for terminating an evolutionary process. If the fitness value of the best individual is greater than some threshold or if the evolution has been run for the specified maximum number of generations, the evolutionary process will be stopped. If one of these two conditions is satisfied, the best individual will be taken as the improved design. Otherwise the GP operations are used to obtain a new population and the evolutionary process will start again from the beginning.

Constraints may also be incorporated into this optimization scheme, as required. For example, the conventional method of using a Lagrange multiplier to specify a constraint within the optimizing cost function may be extended to the fitness function in a direct manner. It can also be used with the fitness function. Specifically, the fitness function with a constraint may be expressed as:

$$F(x, y, G) = f(x, y) + \lambda h(x, y) \quad (2.19)$$

In this equation, F is the fitness function, h is the constraint, and λ is a Lagrange multiplier.

In the conventional approach of GP, the constraints have to be incorporated into the fitness function. An alternative approach would be to introduce constraints through a “supervisor” during the evolution of the solution. As an example, the supervisor may examine a population and eliminate those members that do not satisfy a constraint.

Chapter 3 Simulation Results

The integrated methodology of bond graphs and genetic algorithms, as developed in the present thesis, is now evaluated through computer simulation. The approach is applied in examples of in mechanical and electrical domains. Computer simulations are carried out and the obtained results of evolved systems are discussed.

3.1 An Example in Mechanical Domain

For the mechanical domain, a mass, spring and damper system with a given transfer function is identified. The embryo model for this example is shown in Figure 3.1. This is the starting point of the design evolution. The input to this system is the force applied to the mass m_1 and the output is the velocity of mass m_2 . The objective of this example is to evolve a system having the transfer function:

$$\frac{V(s)}{F(s)} = \frac{4s+40}{s^3+40s^2+400s} \quad (3.1)$$

The fitness values of the individuals are calculated using the absolute error of the actual frequency response magnitude of the candidates and the required frequency response magnitude. For this, 15000 samples were taken in the logarithmic scale in the range of 1- 10^4 rad/s. Finally, the fitness is calculated using:

$$f = \frac{1}{1+E[|error|]} \quad (3.2)$$

Here $E[|error|]$ is the expected value of the absolute error between the magnitudes of the two frequency response functions, as indicated in Figure 3.2.

Figure 3.1: Embryo for the mass spring damper system.

Figure 3.2: Absolute error of the actual and desired frequency response functions as used in fitness calculation.

For this simulation a population size of 60 was used and the final solution was obtained after 20 generations. The final evolved system is shown in Figure 3.3a and the fitness of each generation during evolution is shown in Figure 3.3b.

Figure 3.3: Evolved system and the fitness improvement during evolution for the mass spring damper system: (a) evolved system; (b) fitness evolution.

3.2 Examples of Electrical Systems

The methodology developed in the thesis is now applied to evolve different types of electrical filters. There exists work that has used bond graphs and genetic programming to design passive low-pass filters and passive high-pass filters [Fan et al., 2004; Hu et al., 2005]. In the present work these developments are further improved and extended to design a band-stop filter and a band-pass filter.

3.2.1 Band-stop Filter

In this simulation the objective is to evolve a band-stop filter with a stop band of 5×10^4 - 10^5 rad/s. As the desired system a second order Butterworth approximation of a band pass filter is used. Fitness calculation is done in the same manner as for the mass-spring-damper system except that a frequency range of 1 - 10^9 rad/s is used with 100k equally

spaced samples in a logarithmic scale. Figure 3.4a shows the embryo model used in the beginning of the evolution. The embryo contains two resistors of $100\ \Omega$ and $10^6\ \text{M}\Omega$ and a voltage source. The output is measured across the $10^6\ \text{M}\Omega$ resistor. The evolved solution is given in Figure 3.4b and its frequency response is shown in Figure 3.4c.

Figure 3.4: Band stop filter evolution, (c) frequency response of the evolved filter.

3.2.2 Band-pass Filter

In this example a band-pass filter having a pass band of $5 \times 10^4 - 10^5$ rad/s is evolved. The fitness calculation is done in the same way as the evolution of the band-stop filter, with a second order Butterworth filter as the specification. The exact transfer function for this band-pass filter is

$$H(s) = \frac{5 \times 10^4 s}{s^2 + 5 \times 10^4 s + 5 \times 10^9} \quad (3.3)$$

The frequency response of this transfer function is shown in Figure 3.5. The embryo for this simulation (Figure 3.6a) contains only a voltage source. The evolved final solution is shown in Figure 3.6b.

Figure 3.5: Desired frequency response function of a band pass filter with 2nd order butterworth approximation.

Figure 3.6: Band-pass filter evolution.

3.2.3 Drawbacks in Fitness Calculation

During the evolution of the filters there were cases where the fitness stayed at the same level without improvement. The fitness function highly depends on the objective. In a filter design problem it depends on the frequency range, type of filter, transition band, and so on. Figure 3.7 shows two intermediate systems that were generated during the band-pass filter evolution. System A in Figure 3.7a has an exact topology for a band pass filter while system B in Figure 3.7b has a different topology. Figures 3.8a and 3.8b give the frequency response functions of these two systems. They show that system A has a wider pass-band compared to the desired pass band of $5 \times 10^4 - 10^5$ rad/s. If the fitness values are calculated using the method given in [Fan et al., 2004], where 100 uniformly sampled points are used to calculate the absolute error in the two frequency responses, as in Equation 3.2, it will give the values in Table 3.1. This clearly shows that even up to a range of 10^6 the method of Fan et al. will give a higher fitness values for system B. But ideally we would like to have a better fitness value for system A, since only an improvement in the parameter values is needed for that system. In this kind of situations the evolutions converge to undesirable systems, degrading the performance of the methodology.

Figure 3.7: Two intermediate solutions during band-pass filter evolution.

Table 3.1: Fitness values for systems A and B using the method in [Fan et al., 2004].

Frequency Range (krad/s)	Fitness	
	System A	System B
0-200	0.1074	0.2303
0-500	0.0686	0.1167
0-1000	0.0505	0.0742

(a) System A

(b) System B

Figure 3.8: Frequency response for systems A and B in figure 3.7.

On the other hand making the fitness function too strict will also affect the performance. Often it is good to have a diverse set of systems in a different range of fitness values, which will help the genetic program evolve to a good final solution. These trade-offs have to be considered in establishing a fitness function for this type of problem.

Even though it is difficult to define a fitness function that will have a better fitness for system A, due its wider pass-band, we consider different fitness functions to establish a criterion that will generate more consistent results. First, a weighting method is used for calculating the error between the two frequency responses. Equal weights were given for

the transition frequencies and a higher weight was given for the pass-band such that the sum of the weights was equal to one (see Figure 3.9). With these weightings, the fitness of the frequency response (f_{fr}) is calculated according to:

$$f_{fr} = 0.25Er_1 + 0.5Er_2 + 0.25Er_3 \quad (3.4)$$

Here Er_1 , Er_2 and Er_3 are the error between the desired frequency response and the candidate frequency response in the ranges, $1 - 5 \times 10^4$ rad/s, $5 \times 10^4 - 10^5$ rad/s and 10^5 rad/s to the upper limit of the frequency range of interest, respectively.

Figure 3.9: Weights for frequency response error in different frequency ranges.

Another characteristic of these evolutions is that they tend to evolve arbitrarily for complex system. As a way to control the complexity of the evolved systems, a fitness based on the system order is introduced. For this, a Gaussian distribution is used as a function of system order. This will give a low fitness for a system having too high or too low order. The fitness of the system order (f_{or}) is calculated according to:

$$f_{or} = e^{\frac{-(n-\mu)^2}{2\sigma^2}} \quad (3.5)$$

Here n is the order of the candidate system, and μ and σ are the mean and the standard deviation. A band-pass filter should beat least a second order system. A mean of 2 and a standard deviation of 1 are used for all the simulations.

With these two fitness criteria the total fitness is calculated using 80% of frequency fitness and 20% of order fitness. Since GP introduces some random behavior, a method that works for one case may not work all the time. Therefore, for these simulations each fitness calculation method is tried five times in order to identify the reliability. For all the following simulations the GP parameters are set at the following values:

- Population size - 100
- Maximum number of generations - 200
- Fitness threshold - 0.95
- Repetition rate - 10%
- Junction crossover rate - 20%
- Elements crossover rate - 20%
- Arithmetic crossover rate - 30%
- Mutation rate - 20%

First, this fitness criterion is used with a logarithmic scale for sampling the frequency response. In these simulations, two frequency ranges are used. They are $1-10^6$ rad/s and $1-10^7$ rad/s. For both of these ranges, simulation experiments are carried out using 100, 1000, 10k and 100k samples (n_s).

Figure 3.10 shows the evolutions for different resolutions in a logarithmic frequency scale for the range $1-10^7$ rad/s. In the frequency range $1-10^7$ rad/s range, the evolved result improved as the number of samples was increased, and with 100k samples all the 5 runs attained the final fitness threshold as shown in Figure 3.10.

Figure 3.10: Fitness calculations using a logarithmic scale with different number of samples (n_s) in the range of $1-10^7$ rad/s

In the frequency range $1-10^6$ rad/s, regardless of the number of samples, it was unable to achieve the required fitness value in all five occasions of simulation. Figure 3.11 shows the 5 evolutions for the $1-10^6$ rad/s range and using 100k samples.

Figure 3.11: Results for logarithmic scale for the range $1\text{-}10^6$ rad/s with 100k samples.

Even though using a higher number of samples produced better results, the evolved systems were not perfect. Figure 3.12 shows an evolved system using 100k samples for the frequency range $1\text{-}10^7$ rad/s in the logarithmic scale.

Figure 3.12: Evolved band-pass filter and FRF using log scale with 100k samples in frequency range $1\text{-}10^7$ rad/s; (a) circuit; (b) frequency response.

The transfer function of this system with the parameter values $R_1 = 6.7 \text{ k}\Omega$, $R_2 = 8.5 \text{ M}\Omega$, $R_3 = 0.58 \Omega$, $L_1 = 0.0653 \text{ H}$ and $C_1 = 3.067 \text{ pF}$ is given by:

$$H(s) = \frac{s(s+5.609 \times 10^8)}{s^2 + 4.87 \times 10^4 s + 4.993 \times 10^9} \quad (3.6)$$

The frequency response of this system which is given in Figure 3.12b does not represent an accurate 2nd order Butterworth band-pass filter as shown in Figure 3.5. By observing the transfer function of this system it is clear that there is a zero around $5.6 \times 10^8 \text{ rad/s}$ which causes the deviation of the response from the required function. Therefore, despite using a higher number of samples from a higher frequency range, the identified solution was not accurate. To suppress very high frequencies this fitness criterion may not work. Another reason for this result can be that in a logarithmic scale the high frequencies are sampled at larger intervals. That may not capture the fine variations at higher frequencies. As a solution, the frequency range may be reduced to increase the resolution in upper part of the frequency range. But as shown previously, reduced frequency range will also hinder the performance of the evolution. To overcome this problem, a linear scale is used now, with the same number of samples (100k).

On comparing the error calculated using Equation 3.4 for a linear scale from $1 - 9 \times 10^5 \text{ rad/s}$ and logarithmic scale from $1 - 10^7 \text{ rad/s}$, it was found that the linear scale gave a higher error value for the system in Figure 3.12. As clear from Equation 3.2, a higher error gives a lower fitness value. The errors with respect to the desired response using a linear scale and a logarithmic scale are 5531.4 and 5451.4, respectively. This gives an error difference of 80 which supported the use of a linear scale, since it gave a higher error for the undesired system.

Linear Resolution

For the linear scale, sampling intervals of 10 rad/s and 100 rad/s were used. The frequency range was set to $1 - 9 \times 10^5 \text{ rad/s}$. Simulations were carried out 5 times for

both sampling intervals. Figure 3.13 shows an evolved system using the sampling interval 100 rad/s.

Figure 3.13: An evolved system using linear scale with 100 rad/s sampling period: (a) circuit; (b) frequency response.

Even though the frequency response of this system matches that of the desired system around the pass band, low frequency behavior deviates from the desired function. This shows that, despite using a linear scale, the frequency response of the evolved system may not exactly match the desired response when a sampling interval of 100 rad/s is used. Because of this result, a sampling interval of 10 rad/s is used in different frequency ranges.

Figure 3.14 shows 5 evolutions for the frequency ranges, $1-3 \times 10^5$ rad/s, $1-6 \times 10^5$ rad/s and $1-9 \times 10^5$ rad/s. In these simulations only the frequency range $1-9 \times 10^5$ rad/s was able to attain the required fitness threshold for all five runs. Therefore, a sampling interval of 10 rad/s with a frequency range from $1-9 \times 10^5$ rad/s would be a good choice for the band-pass filter design problem.

(a) $1-3 \times 10^5$ rad/s

(b) $1-6 \times 10^5$ rad/s

(c) $1-9 \times 10^5$ rad/s

Figure 3.14: Results using linear scale with different frequency ranges at sampling period 10 rad/s.

Figure 3.15 shows the two different topologies evolved when using the frequency range $1-9 \times 10^5$ rad/s and the sampling interval 10 rad/s. Even being band-pass filters, these topologies are much simpler than the high-pass filters developed in [Fan et al., 2004] and the low-pass filter developed in [Hu et al., 2005]. This is due to the fact that the system order has been introduced to the fitness function, with the objective of lowering (optimizing) it as well.

Figure 3.15: Two different topologies obtained for evolution of band-pass filter using frequency range $1\text{-}9 \times 10^5$ rad/s and sampling interval 10 rad/s: (a) an evolved circuit; (b) another evolved circuit.

3.3 Identification of a System with Desired Eigenvalues

This design evolution methodology, as developed in this thesis, is now applied for obtaining a system with a specified set of eigenvalues. In the present example three eigenvalues are specified as:

$$\begin{bmatrix} -1000 \\ -100 + 200j \\ -100 - 200j \end{bmatrix}$$

For the fitness calculation, a scaled difference between the real and imaginary parts of the eigenvalues is used according to:

$$E_{im} = \frac{|im(\lambda_d) - im(\lambda_a)|}{\alpha(n)(1 + |im(\lambda_d)|)} \quad (3.7a)$$

$$f_{im} = \frac{1}{1 + E_{im}} \quad (3.7b)$$

$$E_{re} = \frac{|re(\lambda_d) - re(\lambda_a)|}{\alpha(n)(1 + |re(\lambda_d)|)} \quad (3.7c)$$

$$f_{re} = \frac{1}{1 + E_{re}} \quad (3.7d)$$

$$\alpha(n) = 1 - F_m(n) \quad (3.7e)$$

Here λ_d is the desired eigenvalue and λ_a is the actual eigenvalue of the evolved system. The real and imaginary fitness values were calculated for all three eigenvalues using equal weights. In order to obtain a 3rd order system, the Gaussian function of equation 3.5 is used, as explained in Section 3.2.3. Note that F_m is the maximum fitness value in the n^{th} generation. Therefore, the scaling factor $\alpha(n)$ decreases as the fitness value increases. The role of this scaling factor is to give a reasonable fitness for less accurate systems at the start of an evolution and obtain more accurate results towards the end of the evolution. For example, if the fitness value is close to 1, which is the maximum attainable fitness value, the denominators in equations 3.7a and 3.7c will decrease, giving higher E_{im} and E_{re} . But for the generations with low fitness values, the effect of the denominator is greater, giving low E_{im} , E_{re} and eventually a higher fitness value.

Figure 3.16a shows the bond graph of the evolved system for an electrical circuit. The fluctuating behavior of the fitness improvement in Figure 3.16b is due to the adaptive normalizing factor $\alpha(n)$. In this case a solution that gives a particular fitness level is possible to reduced with updated $\alpha(n)$ in the next generation.

Figure 3.16: Identifying a system with specified eigenvalues: (a) circuit bond graph, (b) fitness evolution.

Chapter 4 Experimental Validation

On obtaining successful simulation results from the proposed methodology, an experimental validation is carried out. The industrial machine for fish head cutting (the Iron Butcher) which is available in the Industrial Automation Laboratory is used for this purpose. The Iron Butcher (IB) is an industrial machine which has been developed in our laboratory and transferred to the fish processing industry. It cuts the head of fish resulting in optimal meat recovery. The sensing and feedback capabilities of this machine improve the efficiency and quality of fish processing [de Silva, 1992] compared to conventional fish processing techniques. Moreover, it is an ideal industrial machine for testing this type of methodologies because of its multi-domain nature. It consists of subsystems from different domains including a hydraulic positioning system, electromechanical conveyor system, and a pneumatic system. Past researchers have developed condition monitoring systems and health monitoring systems for the Iron Butcher which are useful for the evolutionary design framework presented in Chapter 1 [Lang and de Silva, 2008; Raman and de Silva, 2009; Razavi and de Silva, 2010].

Figure 4.1 shows a picture of the Iron Butcher, with some of its subsystems indicated. The main subsystems of the Iron Butcher are:

- Conveyor System - This subsystem carries the fish from the feeding point towards the cutter. This is driven by an ac induction motor and a linkage system. Indexing pins are used to stabilize the fish on the conveyor table.
- Vision System - Camera mounted near the cutter blade captures the fish and a vision algorithm computes the optimal location of the gill of the fish to be cut.
- Hydraulic Actuator - Using the data generated by the vision system, the hydraulic actuators position the cutter assembly accordingly.
- Pneumatic System - The cutter blade and fish stabilizing pins are operated through pneumatic actuators.

In addition to these subsystems there are numerous sensors such as limit switches and pressure sensors for the smooth and automated operation of the machine.

Figure 4.1: The iron butcher.

4.1 Experimental Setup

The conveyor subsystem is used in the experimental verification of the developed methodology of design evolution. In operating the machine, a user places the fish on the conveyor table and a set of indexing pins pushes the fish towards the cutter. These indexing pins are moved by a linkage system as shown in Figure 4.2. The linkages are driven by an ac induction motor coupled through a gearbox and a disc. The angular velocity of the induction motor can be controlled by the variable frequency drive and it can also be wired to sense the actual speed of the motor.

A schematic diagram of this linkage system is shown in Figure 4.3. There, the three links are denoted by AD , DE and BC . For the conducted experiment the acceleration at point E is measured as the output.

Figure 4.2: Conveyor system of the iron butcher.

The actual speed of the ac motor is considered as the input to the system. A data acquisition system manufactured by National Instruments (NI) is used to generate the signals required for the variable frequency drive and to acquire the acceleration and motor speed.

The complete experimental verification procedure is given by the flow chart in Figure 4.4. First, a linearized bond graph model of the conveyor system of the fish cutting machine is developed. The parameter values of the BG model are then identified using genetic programming. Then the BG-GP methodology, as developed in the present work, is used to evolve an improved design for the poorly performing segment as suggested by the machine health monitoring system. Finally, the conveyor system is physically modified according to the improved model (improved design) generated by the BG-GP system. Actual data from the modified physical system is used to verify the performance of the improved system.

Figure 4.3: Schematic diagram of the linkage.

Figure 4.4: Experimental validation procedure.

4.1.1 Bond Graph Model

The Iron Butcher is a nonlinear system. Specifically, in the linkage unit, the link velocities depend nonlinearly on the link angles. Figure 4.5 shows the schematic model of the nonlinear linkage system. The motor is modeled as a velocity source (S_f) and the all the other elements are modeled as lumped parameter models.

Figure 4.5: Bond graph model of the nonlinear conveyor system.

The transformer coefficient $f_1(\theta_1, \theta_2, \alpha)$ relates the angular velocity of the disc to the linear velocity of point B and $f_2(\theta_1, \theta_2, \alpha)$ relates the linear velocities of points C and E. The lumped parameters are:

- K_1 - Stiffness of the shaft that couples the gearbox and disc
- K_2 - Stiffness of BC and DE
- J_G - Moment of inertia of the gear
- J_D - Moment of inertia of the disc
- M - Equivalent mass of AD and BC at C
- M_L - Equivalent mass of table and DE at E
- b - Equivalent viscous damping constant at E
- r_{GB} - Gearbox ratio

For the experimental validation, the system is linearized at a point where link AD is oriented vertically upward. Small oscillations around this point are considered during this experiment. This makes the links ED and CB almost horizontal, and the transformer coefficients $f_1(\theta_1, \theta_2, \alpha)$ and $f_2(\theta_1, \theta_2, \alpha)$ become r_w and k_{AD} , where r_w is the disc radius and k_{AD} is the ratio between the linear velocities at points C and E. Since link AD has a pure rotational movement, one has

$$k_{AD} = \frac{AC}{AD}. \quad (4.1)$$

The linearized bond graph model of the conveyor system is shown in Figure 4.7.

4.1.2 Equations of Motion

The equations of motion of the conveyor system and the associated bond graph model are derived and verified now. Consider a case where the table is pushed towards the left as in Figure 4.6. Since the motor and gearbox are directly coupled, the angular velocities at the input of the gear box (ω_i) and the motor (ω_m) are the same.

Figure 4.6: An instance of the table moving to left (f_4 and f_6 are the reaction forces exerted on link AD by CB and ED).

Note that ω_m and τ_m are the angular velocity and torque of the motor. One has:

$$J_g \dot{\omega}_m = \tau_g \quad (4.2)$$

Motor torque will drive the gear inertia and supply the input torque (τ_i) to the gear box, which satisfies

$$\tau_m = \tau_g + \tau_i \quad (4.3)$$

The output torque (τ_o) and the angular velocity (ω_o) of the gear box are given by

$$\omega_o = r_g \omega_m \quad (4.4)$$

$$\tau_o = \frac{1}{r_g} \tau_i \quad (4.5)$$

This output torque is applied to the flexible shaft (K_1) which couples the disc and the gearbox. One may write:

$$\tau_{k_1} = \tau_o \quad (4.6)$$

and

$$\frac{d\tau_{k_1}}{dt} = K_1(\omega_o - \omega_d) \quad (4.7)$$

where ω_d is the angular velocity of the disc.

There are two torques on the disc: the reaction from link CB (τ_2) and the torque applied from the shaft. Hence,

$$J_d \ddot{\omega}_d = \tau_{k_1} - \tau_2 \quad (4.8)$$

$$\tau_2 = r_w f_3 \quad (4.9)$$

$$\omega_d = \frac{1}{r_w} v_3 \quad (4.10)$$

where J_d is the disc inertia and v_3 is the linear velocity at point B. For the shaft flexibility (K_2), one has

$$\frac{df_3}{dt} = K_2(v_3 - v_4) \quad (4.11)$$

where v_4 is the linear velocity at point C and f_3 is the force through the link CB. For the equilibrium of CB link,

$$f_3 = f_4 \quad (4.12)$$

For link AC, taking moments about A, for a small angular displacement θ , the following equation may be written:

$$J_{AD} \ddot{\theta} = f_4 \cdot AC - f_6 \cdot AD \quad (4.13)$$

By rearranging, one obtains

$$\frac{J_{AD}}{AC^2} \cdot (AC \ddot{\theta}) + f_6 \frac{AD}{AC} = f_4 \quad (4.14)$$

Taking $\frac{J_{AD}}{AC^2}$ as the equivalent link mass (M) at point C and $AC \ddot{\theta}$ as the linear acceleration $\dot{v}_M$, one has from (4.13):

$$M \dot{v}_m + f_6 k_{AD} = f_4 \quad (4.15)$$

where $k_{AD} = \frac{AD}{AC}$. Also,

$$\frac{v_6}{AD} = \frac{v_4}{AC} \quad (4.16)$$

$$v_6 = v_4 k_{AD} \quad (4.17)$$

For the table mass (M_L) one may write

$$M_L \dot{v}_l = f_6 - f_b \quad (4.18)$$

where v_L is the table velocity and f_b is the viscous friction. Also,

$$f_b = b v_l \quad (4.19)$$

4.1.3 BG Model

The bond graph model of the system is shown in Figure 4.7.

Figure 4.7: Bond graph model.

From the BG model it is seen that for the common flow junction, the motor and gear box connected. Thus,

$$\omega_m = \omega_i = \omega_g \quad (4.20)$$

For the gear inertia:

$$J_g \dot{\omega}_g = \tau_g \quad (4.21)$$

From 4.20:

$$J_g \dot{\omega}_m = \tau_g \quad (4.22)$$

=> Equation 4.2.

Using continuity for this common flow junction, one has

$$\tau_m = \tau_g + \tau_i \quad (4.23)$$

=>Equation 4.3.

For the first transformer, one may write

$$\omega_o = r_g \omega_i \quad (4.24)$$

$$\tau_o = \frac{1}{r_g} \tau_i \quad (4.25)$$

For the common effort junction to which K_I is attached, the continuity equations may be written as

$$\omega_o = \omega_{k_1} + \omega_1 \quad (4.26)$$

$$\Rightarrow \omega_{k_1} = \omega_o - \omega_1$$

$$\tau_o = \tau_{k_1} = \tau_1 \quad (4.27)$$

For K_I :

$$\frac{d\tau_{k_1}}{dt} = K_1 \omega_{k_1} \quad (4.28)$$

For the common flow to which J_d is attached, one has

$$\tau_1 = \tau_d + \tau_2 \quad (4.29)$$

$$\Rightarrow \tau_d = \tau_1 - \tau_2$$

$$\omega_1 = \omega_d = \omega_2 \quad (4.30)$$

From 4.26, 4.28, and 4.30, one obtains

$$\frac{d\tau_{k_1}}{dt} = K_1(\omega_o - \omega_d) \quad (4.31)$$

$\Rightarrow$ Equation 4.7.

For J_d :

$$J_d \dot{\omega}_d = \tau_d \quad (4.32)$$

(4.27), (4.32)&(4.29) $\Rightarrow$

$$J_d \dot{\omega}_d = \tau_{k_1} - \tau_2 \quad (4.33)$$

$\Rightarrow$ Equation 4.8.

For the transformer coefficient r_w one may write

$$f_3 = \frac{1}{r_w} \tau_2 \quad (4.34)$$

$$v_3 = r_w \omega_2 \quad (4.35)$$

For the common effort junction to which K_2 is attached, applying the continuity equations, one obtains

$$f_3 = f_{k_2} = f_4 \quad (4.36)$$

$$v_3 = v_{k_2} + v_4 \Rightarrow$$

$$v_{k_2} = v_3 - v_4 \quad (4.37)$$

For K_2 :

$$\frac{df_{k_2}}{dt} = K_2 v_{k_2} \quad (4.38)$$

$$(4.36), (4.37) \Rightarrow \frac{df_3}{dt} = K_2 (v_3 - v_4) \quad (4.39)$$

$\Rightarrow$ Equation 4.11.

For the common flow junction to which link mass M is attached, applying the continuity equations, one obtains

$$v_4 = v_m = v_5 \quad (4.40)$$

$$f_4 = f_m + f_5 \Rightarrow$$

$$f_m = f_4 - f_5 \quad (4.41)$$

For M :

$$f_m = M \dot{v}_m \quad (4.42)$$

For the transformer with coefficient k_{AD} ,

$$v_6 = v_5 k_{AD} \quad (4.43)$$

$$f_6 = \frac{1}{k_{AD}} f_5 \Rightarrow$$

$$f_5 = k_{AD} f_6 \quad (4.44)$$

From equations 4.41 and 4.42:

$$f_4 - f_5 = M \dot{v}_m \quad (4.45)$$

$$(4.44) \Rightarrow f_4 - k_{AD} f_6 = M \dot{v}_m \Rightarrow$$

$$M \dot{v}_m + k_{AD} f_6 = f_4 \quad (4.46)$$

$\Rightarrow$ Equation 4.15.

For the common flow junction to which ML is attached, applying the continuity equations give:

$$f_6 = f_l + f_b \Rightarrow$$

$$f_l = f_6 - f_b \quad (4.47)$$

$$v_6 = v_l = v_b \quad (4.48)$$

For load M_l :

$$f_l = M_l \dot{v}_l \quad (4.49)$$

(4.47) =>

$$f_6 - f_b = M_l \dot{v}_l \quad (4.50)$$

=> Equation 4.18.

For b :

$$f_b = v_b b \quad (4.51)$$

(4.48) =>

$$f_b = v_l b \quad (4.52)$$

=> Equation 4.19.

State-Space Model

From these equations, the following state-space mode is obtained.

State vector,

$$\mathbf{X} = \begin{bmatrix} \tau_{k1} \\ \omega_d \\ f_{k2} \\ v_m \end{bmatrix}$$

State equations,

$$\dot{\tau}_{k1} = K_1 r_g \omega_m - K_1 \omega_d \quad (4.53a)$$

$$\dot{\omega}_d = \frac{1}{J_d} \tau_{k1} - \frac{r_w}{J_d} f_{k2} \quad (4.53b)$$

$$\dot{f}_{k2} = K_2 r_w \omega_d - K_2 v_m \quad (4.53c)$$

$$\dot{v}_m = \frac{-k_{AD}^2 b}{M + k_{AD}^2 M_l} v_m + \frac{1}{M + k_{AD}^2 M_l} f_{k2} \quad (4.53c)$$

The system matrices are:

$$\mathbf{A} = \begin{bmatrix} 0 & -K_1 & 0 & 0 \\ \frac{1}{J_d} & 0 & \frac{r_w}{J_d} & 0 \\ 0 & K_2 r_w & 0 & K_2 \\ 0 & 0 & \frac{1}{M + k_{AD}^2 M_l} & \frac{-k_{AD}^2 b}{M + k_{AD}^2 M_l} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} K_1 r_g \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The acceleration at point E of the system corresponds to $\dot{v}_6$ of the bond graph model. This can be written as,

$$\dot{v}_6 = \dot{v}_m k_{AD} \quad (4.54)$$

This result has used equations 4.40 and 4.43. Therefore the matrices $\mathbf{C}$ and $\mathbf{D}$ of the state-space model are

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & \frac{k_{AD}}{M + k_{AD}^2 M_l} & \frac{-k_{AD}^3}{M + k_{AD}^2 M_l} \end{bmatrix}$$

$$\mathbf{D} = [0]$$

4.1.4 Parameter Identification

After developing the linearized model of the conveyor system, genetic programming is used to identify the model parameters. For this model six parameters have to be identified. These parameters are, K_l , K_2 , J_D , M , M_L and b . The gear ratio (r_g) is 106.8. By measuring the lengths AC and CD , the parameter k_{AD} is calculated as 0.6. The disc radius $r_w = 6$ cm.

For the parameter identification, a sinusoidal velocity profile is used as the input. This is done by sending an equivalent voltage waveform through the NI data acquisition system to the variable frequency drive. Since the internal dynamics of the VFD cannot be determined, the actual velocity of the motor is used as the input. This can be easily done since the motor speed can be acquired from the VFD. For the parameter identification,

the bond graph model is evolved allowing the model parameters to be change during evolution. The actual acceleration of the system is measured using an accelerometer and the bond graph is simulated to obtain the model acceleration. This way, the absolute error between the model acceleration and the actual system acceleration is minimized. Since there are 6 parameters to be identified, three sinusoids at different frequencies are used and the error is minimized for all three responses, simultaneously. The acceleration and the motor speed are sampled at 500 samples per second. The actual motor speed is also used as the input for the simulations. The three frequencies used as the input are 0.2 Hz, 0.21 Hz and 0.22 Hz. The signals are sampled using a low pass filter with a cut off frequency of 1 Hz. The accelerometer signals before and after filtering are shown in Figure 4.8.

Figure 4.8: Unfiltered and filtered acceleration signals.

As the fitness function for the genetic program, the absolute error between the simulated response and the actual response is calculated at the sample points, as in equation 3.2. Equal weights are given for the three errors in calculating the total fitness value. The identified parameter values from this method are found to be:

- $K_1 = 139.08 \text{ Nm/rad}$
- $K_2 = 406.5 \text{ N/m}$
- $M = 2.09 \text{ kg}$

- $M_L = 26.29 \text{ kg}$
- $J_D = 1.01 \text{ kgm}^2$
- $b = 170.9 \text{ Ns/m}$

The actual acceleration and the acceleration obtained from the bond graph model with these parameters are shown in Figure 4.9.

4.2 Design Improvement

For the experimental verification of the developed methodology, a design improvement is made to the conveyor system of the Iron Butcher as prescribed by the BG-GP system. The results are compared with the actual response of the physically modified conveyor system. During the operation of the fish cutting machine, the machine health monitoring system (MHMS) has detected a reduction of speed of the conveyor system. Accordingly, the MHMS suggests a peak acceleration of 9.6 cms^{-2} for a sinusoidal input with an amplitude of 382 rpm and a frequency of 0.2 Hz as a possible solution for this problem. The BG-GP methodology is used in order to evolve a system that is able to generate this required acceleration. For this design evolution, an embryo with the second transformer as a modifiable site is used. This is set as an arithmetic modifiable site so that the k_{AD} value is changed during evolution. This embryo model is shown in Figure 4.10.

Five consecutive peaks are chosen for fitness calculation, as in Figure 4.11. Fitness value is calculated using the absolute error of these five peaks, using

$$fitness = \frac{1}{1 + \sum_{i=1}^5 e_i} \quad (4.55)$$

where e_i is the error at peak i . The evolution is run with a population size of 100 individuals.

(a) 0.20 Hz

(b) 0.21 Hz

(c) 0.22 Hz

Figure 4.9: Acceleration profiles of the actual system and the parameter identified model at different frequencies.

Figure 4.12: Modified linkage as prescribed by the proposed BG-GP methodology.

In order to verify the value for k_{AD} as prescribed by the BG-GP system, the acceleration is again measured after setting k_{AD} to the new value. This acceleration profile is compared with that obtained from the BG model, as shown in Figure 4.13. The horizontal line in Figure 4.13 is the peak acceleration that was prescribed by the MHMS. It is seen that the two acceleration profiles, one from the BG model and the actual acceleration of the modified conveyor system, closely agree with each other. The difference in the first cycle can be attributed to some transients that have not been modeled.

Figure 4.13: Accelerations of the actual system after modification and the evolved BG model.

Figure 4.14 shows the fitness improvement during this evolution, and the evolved GP tree. The developed system of design evolution was able to identify the final solution in 42 generations.

Figure 4.14: (a) Evolved GP tree representing k_{AD} ; (b) fitness improvement during design evolution.

Chapter 5 Conclusion

Modern day engineering design is a complex task involving different engineering domains as well as different socio-economic aspects. Since evolutionary design plays an important role in this type of multi-objective problems, evolutionary design methodology which incorporated bond graph modeling and genetic programming was presented in this thesis. This research was based on a broader framework that consists of a machine health monitoring system to detect design weaknesses of an existing machine, bond graph-based modeling tool to represent the machine during a particular state, a design optimization scheme that uses genetic programming, and a design expert system that supervises the design evolution in order to guide it towards a feasible solution. Only the aspects of bond-graph modeling and genetic programming were treated in the present thesis.

The main objective of this research was to develop aevolutionary design methodology that integrated genetic programming and bond graph modeling, and to verify the effectiveness of this methodology using computer simulations and experimentation of a real world system. The proposed methodology was found to be capable of evolving systems modeled in bond graphs to a desired design that satisfies a required behaviour. Different types of fitness functions (objective functions) were tested for evolutionary optimization. In addition to response error, system complexity was also taken into account in the fitness function of design evolution. Engineering systems in both mechanical and electrical domains were used to test the effectiveness of the developed method. Both computer simulation and laboratory experimentation were used for this purpose. The experimental verification was done using the Iron Butcher, and industrial machine for fish processing, which had been developed at the Industrial Automation Laboratory of the University of British Columbia and used in the fish processing industry. The results showed the ability of the developed method to successfully improve the designs of these engineering systems.

Even though the work in this thesis was based on linear lumped parameter models, it is extendable to other systems that include nonlinearities such as Coulomb

friction, which may be represented by bond graphs. Also, models of mechanics of materials may be incorporated for representing material properties including strength. These special elements may be incorporated using construction functions in bond graph models.

5.1 Summary and Significance of the Contributions

The main contributions and their significance of the work presented in this thesis can be summarized as follows:

- Development of an evolutionary design methodology which integrated system modeling using bond graphs and design optimization using genetic programming. A particular design of an engineering system was represented by a bond graph which in turn was represented as a tree structure that can be optimally evolved using genetic programming to a desired system. Developed a methodology for automatically extracting the state space model from a bond graph and simulating them to obtain their responses. The system complexity also was incorporated in the fitness, which is optimized during design evolution. A mass-spring-damper mechanical system and several types of passive electrical filters were evolved using the developed method, which generated designs of less complexity compared to the results obtained by others in previous work.
- The fitness function used in BG-GP-based evolutionary schemes has not received sufficient attention in previous work. The present work showed some shortcomings in the fitness calculation and proposed improvements for the fitness function. It was shown that more reliable and accurate results can be generated through proper fitness functions.
- Since most of the past research on BG-GP methodologies was done using computer simulations, going one step further, the design evolution methodology as developed in the present thesis was verified using a real world physical system.

Specifically, a modification was done to an industrial machine according to the prescription from the developed method, and the result was verified to be satisfactory by data from actual modified system and the corresponding model. A linear bond graph model of an industrial fish processing machine was also developed in this process and its parameter values were identified using genetic programming.

5.2 Possible Future Directions

The BG-GP methodology can be further improved by incorporating more elements. Though the present research has used the basic components R , I , C elements, transformers and gyrators, one can use user-defined elements like several components built into one element, nonlinear elements, and so on. Improvements to the bond graph model of the Iron Butcher are desirable, and it will benefit from these additional elements and will allow the use of a wide range of inputs. Later non-linear simulations should be added to obtain more accurate results.

Genetic programming can be improved in the context of bond graph evolution. As presented in Chapter 2, some of the genetic operations have been done only with matching nodes of the GP tree. Similarly, different types of intelligence can be incorporated into genetic programming. This may include having parameter values that are realistic for a specific problem, using the component values that are commercially available, and adding the component cost into the fitness function.

Despite the work presented in Chapter 3 for fitness improvement, still there is room for improving the fitness function. An important observation that was made during filter design was, having a fitness criterion for the topology would increase the evolution efficiency to a great extent. Two main aspects of system design are identification of the system topology and identification of the parameter values. Out of them, identification of the topology is the hardest part. Therefore, an improved fitness criterion should be developed to specify the required topology for different responses. Furthermore, since the

fitness function is more subjective, one may investigate different aspects to be considered in identifying various types of systems in different engineering domains.

Bibliography

Modelica software. <https://modelica.org>.

T. Back, U. Hammel, and H.-P. Schwefel. Evolutionary computation: comments on the history and current state. *IEEE Transactions on Evolutionary Computation*, 1(1):3 –17, Apr 1997.

Y. Bar-Yam. When systems engineering fails-toward complex systems engineering. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 2021 – 2028 vol.2, Oct. 2003.

S. Behbahani. Practical and analytical studies on the development of formal evaluation and design methodologies for mechatronic systems. PhD thesis, University of British Columbia, 2007.

S. Behbahani and C.W. de Silva, Mechatronic Design Quotient (MDQ) as the Basis of a New Multi-Criteria Mechatronic Design Methodology. *IEEE/ASME Transactions on Mechatronics*, Vol. 12, No. 2, pp. 227- 232, 2007

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 Transactions on Mechatronics*, 13(1):14 –21, Feb. 2008.

W. Borutzky. Bond graphs and object-oriented modelling - a comparison. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 216(1):21–33, 2002.

W. Borutzky. Bond Graph Methodology. Development and Analysis of Multidisciplinary Dynamic System Models. Springer, 2010.

J. Broenink. Object-oriented modeling with bond graphs and modelica. In *Proceedings of the International Conference on Bond Graph Modeling and Simulation*, pages 163–168, 1999.

C. Coello. An updated survey of evolutionary multi objective optimization techniques: state of the art and future trends. In *Proceedings of the 1999 Congress on Evolutionary Computation*, volume 1, pg. 3, vol. (xxxvii+2348), 1999.

C. A. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen. Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, 2nd edition, 2007.

C.W. de Silva. Research Laboratory for Fish Processing Automation. *International Journal of Robotics and Computer-Integrated Manufacturing*, Vol. 9, No. 1, pp. 49-60, 1992.

C. W. de Silva. *Mechatronics—An Integrated Approach*. CRC Press, Boca Raton, FL, 2005.

C.W. de Silva, C.W., A Paradigm Shift in Machine Health Monitoring: A Unified System Framework Including Intelligent Supervisory Control and On-Line Design Evolution with Networked Remote Operation. *Tier 1 Canada Research Chair in Mechatronics and Industrial Automation*, the University of British Columbia, Vancouver, BC, Canada, 2008.

C. W. de Silva. *Modeling and Control of Engineering Systems*. CRC Press, Boca Raton, FL, 2009.

C. W. de Silva. *Mechatronics—A Foundation Course*. CRC Press, Boca Raton, FL, 2010.

Z. Fan, K. Seo, J. Hu, E. D. Goodman, and R. C. Rosenberg. A novel evolutionary engineering design approach for mixed-domain systems. *Engineering Optimization*, 36(2):127–147, 2004.

F. A. Firestone. A new analogy between mechanical and electrical systems. *Journal of the Acoustical Society of America*, 1933.

L. B. Gamage and C. W. de Silva. A system framework with online monitoring and evaluation for design evolution of engineering systems. *Journal of Computing and Information Science in Engineering*, 10(3):034501, 2010.

P. Gawthrop and G. Bevan. Bond-graph modeling. *Control Systems, IEEE*, 27(2):24–45, april 2007.

J. Grimbleby. Automatic analogue network synthesis using genetic algorithms. In *Genetic Algorithms in Engineering Systems: Innovations and Applications*, 1995. GALESIA. First International Conference on (Conf. Publ. No. 414), pages 53–58, sep 1995.

J. H. Holland. *Adaption in Natural and Artificial Systems*. University of Michigan Press, 1975.

- J. Hu, X. Zhong, and E. D. Goodman. Open-ended robust design of analog filters using genetic programming. In *Proceedings of the 2005 conference on Genetic and evolutionary computation, GECCO '05*, pages 1619–1626, New York, NY, USA, 2005. ACM.ISBN 1-59593-010-8.
- J. Hu, E. D. Goodman, S. Li, and R. Rosenberg. Automated synthesis of mechanical vibration absorbers using genetic programming. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 22(03):207–217, 2008.
- F.O. Karray and C.W. de Silva. *Soft Computing and Intelligent Systems Design—Theory, Tools, and Applications*, Addison Wesley, New York, NY, 2004
- D. C. Karnopp, D. L. Margolis, and R. Rosenberg. *System Dynamics, Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons, Inc, 4th edition, 2006.
- J. Koza, I. Bennett, F.H., D. Andre, M. Keane, and F. Dunlap. Automated synthesis of analog electrical circuits by means of genetic programming. *IEEE Transactions on Evolutionary Computation*, 1(2):109–128, Jul 1997.
- J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.
- J. R. Koza. Survey of genetic algorithms and genetic programming. In *WESCON'95. Conference record. 'Microelectronics Communications Technology Producing Quality Products Mobile and Portable Power Emerging Technologies'*, page 589, Nov 1995.
- J. R. Koza, M. A. Keane, M. J. Streeter, T. P. Adams, and W. J. Lee. Invention and creativity in automated design by means of genetic programming.
- Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 18(03):245–269, 2004.
- K. Krishnaswamy and P. Y. Li. Bond graph based approach to passive teleoperation of a hydraulic backhoe. *Journal of Dynamic Systems, Measurement, and Control*, 128(1):176–185, 2006.

- K. Kristinsson and G. Dumont. Genetic algorithms in system identification. *In Proceedings of the IEEE International Symposium on Intelligent Control*, 1988.,pages 597 –602, Aug 1988.
- H. Lang and C.W. de Silva. Fault Diagnosis of an Industrial Machine through Sensor Fusion. *International Journal of Information Acquisition*, Vol. 5, No. 2, pp. 93-110, June 2008.
- D. L. Margolis. A survey of bond graph modelling for interacting lumped and distributed systems. *Journal of the Franklin Institute*, 319(1-2):125 – 135,1985.
- S. Mason. Feedback theory-some properties of signal flow graphs. *Proceedings of the IRE*, 41(9):1144 –1156, Sept. 1953.
- J. McPhee. Unified modelling theories for the dynamics of multi disciplinary multibody systems. *In Advances in Computational Multibody Systems, volume 2 of Computational Methods in Applied Sciences*, pages 125–154.Springer Netherlands, 2005.
- G. Pahl and W. Beitz. Engineering Design, A Systematic Approach. Springer-Verlag, Heidelberg, 2 edition, 1996. ISBN 3540199179.
- H. M. Paynter. Analysis and Design of Engineering Systems. M.I.T. Press,Cambridge, MA, 1960.
- T. Quarles, A. R. Newton, D. O. Pederson, and Sangiovanni-Vincentelli. SPICE3 Version 3F5 User's Manual, Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, 1994.
- S. Raman and C. W. de Silva. Classifier design for sensor-fault tolerant condition monitoring in an industrial machine. *In Proceedings of the ASME 2009 Dynamic Systems and Control Conference*, number 48920, pages 637–643,2009.
- B. Razavi and C.W. de Silva, “Condition Monitoring in a Hydraulic System of an Industrial Machine Using Unscented Kalman Filter. *International Journal of Information Acquisition*, Vol. 7, No. 3, pp. 177-192, September 2010.
- B. Samarakoon, L. B. Gamage, and C. W. de Silva. Controlled evolution of engineering systems using bond graph and genetic programming. *In Proceedings of the 23rd Canadian Congress of Applied Mechanics*, 2011.

K. A. Smith. Engineering design. *Journal of Engineering Education*, 89, April2000.

C. Sueur and G. Dauphin-Tanguy. Bond-graph approach for structural analysis of mimo linear systems. *Journal of the Franklin Institute*, 328(1):55 – 70, 1991.

E. H. Tay, W. Flowers, and J. Barrus. Automated generation and analysis of dynamic system designs. *Research in Engineering Design*, 10:15–29, 1998.

J. Wang, Z. Fan, J. Terpenney, and E. Goodman. Knowledge interaction with genetic programming in mechatronic systems design using bond graphs. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 35(2):172 – 182,may 2005.

K. Watanabe and K. Izumi. A survey of robotic control systems constructed by using evolutionary computations. In *Proceedings of the 1999 IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 758 –763 vol.2, 1999.