Search for Human Competitive Results in Open Ended Automated Synthesis of a Primordial Mechatronic System

Saheeb Ahmed Kayani
Department of Mechanical Engineering,
College of Electrical and Mechanical Engineering,
National University of Sciences and Technology,
Peshawar Road, Rawalpindi-46000, Pakistan
saheebk@ceme.edu.pk

ABSTRACT

The multi domain nature of a mechatronic system makes it difficult to model using a single modeling technique over the whole system as varying sets of system variables specific to each energy domain are required. Bond-Graphs offer an advanced object oriented modeling and simulation technique. They are domain independent allowing straight forward and efficient model composition, classification and analysis. Bond-Graph model of the mechatronic system can be directly simulated on a digital computer using simulation softwares like 20-Sim© and Modelica© graphically or manipulated mathematically to develop state-space representation using a simplified set of power and energy variables. The simulation scheme can be augmented to synthesize designs for mechatronic systems using genetic programming as a tool for open ended topology or functional design search. This research paper presents results of an experiment developed to combine Bond-Graphs with genetic programming for unified and automated design/synthesis of a simple/primordial multi domain dynamic or mechatronic system.

Categories and Subject Descriptors

I.2.2 [**Artificial Intelligence**]: Automatic Programming; J.2 [**Physical Sciences and Engineering**]: Engineering.

General Terms: Design, Experimentation, Verification.

Keywords: Bond-Graphs, Genetic Programming, Unified/Automated Design, Topology Synthesis, Multi Domain Dynamic or Mechatronic Systems, Physical Design Realization, Dynamic Analysis.

1. INTRODUCTION

Traditionally a mechatronic system has been defined as a multi domain dynamic system combining mechanical, electrical, hydraulic, pneumatic and thermal components. To perform correctly mechatronic systems depend on the interaction of sensors, computers or microcontrollers and actuators. Taking complicated dynamic multi domain systems all the way from concept to prototype requires mathematical models. Such models can include those whose equations the modeler derives directly or develops with software that holds mathematics in the background. To model a mechatronic system all multi domain sub systems must be connected and all non linearities typical of a specific energy domain must be accounted for. To do this a language is needed to describe the different energy domains in communal terms. Using such a language sub models can be connected in an overall system model which can then be simulated on a computer. Interacting physical systems store, transport and dissipate energy among sub systems. Only Bond-Graphs can provide a concise pictorial representation of these interacting dynamic systems down to the topological level [5], [6]. When using Bond-Graphs for mechatronic system representation we can assume that the mechatronic system under consideration is an *n* port mechatronics network with e_l and f_l being system input effort and flow signals and e_n and f_n being system output effort and flow signals respectively [2], [16].

The basic idea of unified and automated design is to replace the role of domain knowledge with the abundant computational resources available to the engineers these days. Genetic programming based simulated evolution techniques are capable of synthesizing designs of arbitrary complexity as the representation of designs is entirely open ended. In [16] a design approach for generating mechatronic systems has been identified by Jiachuan Wang et al. in 2005. According to this methodology for any mechatronic system a start up design is specified at the initial stage. Then Bond-Graph representation is developed and it is transferred to the genetic programming tool which generates initial population, evaluates it according to the fitness function, reconfigures the population and repeats the process until the design criteria are met. The successful conceptual design candidates are transformed into final design again represented as Bond-Graph models. During this whole process information is extensively exchanged with knowledge caches and incorporated in initial and final stages of the design process.

2. IMPLEMENTATION OF THE DESIGN METHODOLOGY

This experimental verification is based on the methodology for unified/automated design of mechatronic systems followed by Kisung Seo et al. in [15] and Jianjun Hu in [4]. A brief summary of the methodology followed for implementing the automated design scheme is included. (a) A Bond-Graph model is specified. (b) First population of genetic programming trees is created. (c) Each individual is evaluated for fitness using fitness function. (d) Genetic programming operations i.e. selection, reproduction, crossover and mutation are performed for each population. (e) Physical design is realized if termination condition of genetic programming run is satisfied. (f) Otherwise the process is repeated starting from fitness evaluation of each individual. In [15] a two step process is employed for evaluation of Bond-Graph models. First each model is analyzed for causality and then state equations are derived identifying whether the system is linear or not. In the next step the fitness of the model is analyzed using fitness criterion. In all previous implementations of the methodology a UNIX based genetic programming software lil-gp 1.01 [17] has been used. For

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07.

the first time Genetic Programming Studio 1.0 [12] is employed in this application which is based on lil-gp 1.01 kernel but offers a visual platform for executing genetic programming code. The code has been written using MS Visual C++ 6.0. The problem implemented is an eigen value design problem from [15]. Two target eigen values $-1\pm 2j$ are selected and a Bond-Graph model of a physical system whose characteristic equation on solving for roots (poles) generates these eigen values is to be evolved. An embryo Bond-Graph model is specified with only one modifiable site highlighted by a dashed oval marking called the write head. The resistors are given same impedance values (R1:250, R2:500) as in [15] to achieve comparable results. The embryo is allowed only one modifiable site to keep the design process and programming simple and less complicated to implement and interpret. The fitness function includes two parameters namely raw fitness and normalized fitness. Raw fitness $Fitness_{Raw}$ is the sum of distances between target eigen values and the nearest solution eigen values after they have been paired.

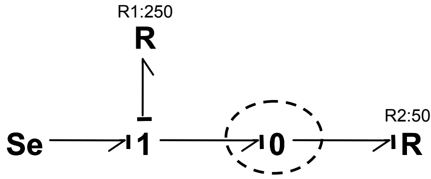

Figure 1. The embryo Bond-Graph model with modifiable site at the zero junction highlighted by dashed oval marking.

Normalized fitness $Fitness_{Norm}$ is calculated according to the relation given as equation 1.

$$Fitness_{Norm} = 0.5 + \frac{1}{1 + Fitness_{Raw}} \tag{1}$$

The developmental or construction procedure for a Bond-Graph model or phenotype is identified by the genetic programming tree or the genotype. Using LISP format of representing genetic programming trees the simulation tool of Genetic Programming Studio 1.0 is used to print long hand versions of such genetic programming trees which need to be simplified for extracting meaningful information. One genetic programming tree represents one individual.

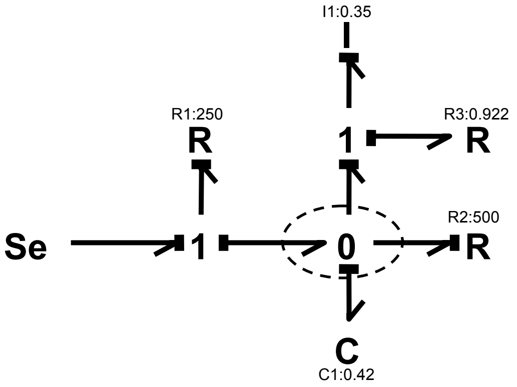

Figure 2. The final simplified Bond-Graph model.

The best solution eigen values compared to target eigen values are included in Table 1 along with average distance error. This table also contains number of *R*, *C*, *I* and junction elements added to the write head. Numerical values of one port elements are also shown. It is to be noted that the eigen values are determined using the *A* matrix of the Bond-Graph model (when state-space equations are written in matrix form assuming the system is linear) containing state variables contributed by the energy storing *C* and *I* elements.

Table 1. Summary of results

Target Eigen Values	
$-1\pm 2j$	
Solution Eigen Values	
$-0.78\pm 1.063j$	
Average Distance Error	
0.961	
Evolved Structure on Write Head	
R Elements	1
C Elements	1
I Elements	1
Junctions	1
Bonds	4
Bond-Graph Element Values	
R Element	0.922
C Element	0.42
I Element	0.35

The values achieved in this experiment are slightly different from the values in [15]. Limitations in writing the code used for generating the Bond-Graph models and tuning of the fitness evaluation process may be the reason which can be removed with more rigorous effort devoted towards problem implementation.

3. DYNAMIC ANALYSIS OF THE EVOLVED SYSTEM

The evolved Bond-Graph model of the physical system is analyzed using 20-Sim© modeling and simulation software. The model contains two energy storing elements I_1 and C_1 therefore it is identified as a second order open loop system with two state variables. The evolved parameters are contained with in the periphery of the dotted square box. The general state-space representation for Bond-Graph model in Figure 4 is given as appearing in equations 3 and 4 [13], [14].

$$\frac{d}{dt}\{X\} = A\{X\} + B\{U\} \tag{2}$$

$$\{Y\} = C\{X\} + D\{U\} \tag{3}$$

In equations 3 and 4 $\{X\}$ is vector of states (momentum P and displacement Q), n is number of states, A is $n\times n$ square matrix, B is $n\times m$ matrix (m is the number of sources), $\{U\}$ is vector of sources (*Se* and *Sf*), $\{Y\}$ is vector of observer states (outputs), l is number of observer outputs, C is $l\times n$ matrix and D is $l\times m$ matrix.

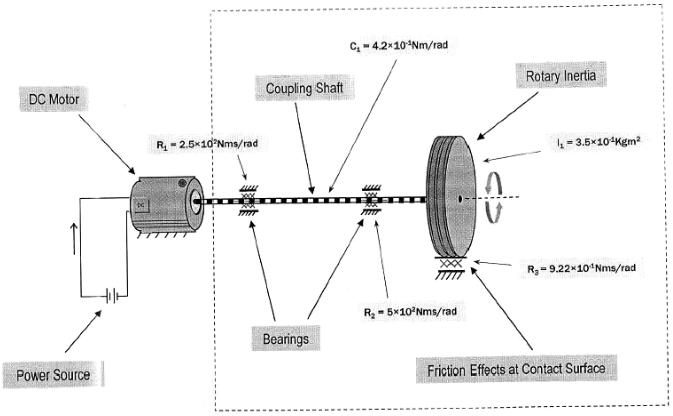
$C_1 = 4.2\times10^{-1}$Nm/rad
Coupling Shaft
Rotary Inertia
DC Motor
$R_1 = 2.5\times10^{2}$Nms/rad
$I_1 = 3.5\times10^{-1}$Kgm^2
DC
$R_3 = 9.22\times10^{-1}$Nms/rad
$R_2 = 5\times10^{2}$Nms/rad
Bearings
Power Source
Friction Effects at Contact Surface

Figure 3. Intuitive physical design realization of the evolved Bond-Graph model.

The poles of the physical system being represented by this Bond-Graph model are determined by calculating eigen values from matrix A in equation 2 using relation $| A - \lambda I | = 0$ where I is identity matrix of order $n\times n$. In due course of the procedure followed for dynamic analysis of the system source of effort *Se* is replaced with a modulated source of effort *MSe* and a motion profile tool provided by the 20-Sim© modeling and simulation software is added to the workspace and connected to *MSe*. Motion profile selected is ramp with unit step as the input or excitation. Output signal is position or the observed output state is displacement *x(t)*. The response of the system to the unit step input is plotted in Figure 4. The values observed from the response curve appear as equations 4-7.

$$\text{Settling Time} = T_s = 5 \text{ s} \tag{4}$$

$$\text{Rise Time} = T_r = 1 \text{ s} \tag{5}$$

$$\text{Peak Time} = T_p = 2 \text{ s} \tag{6}$$

$$\text{Damping Ratio} = \xi = 0.591 \tag{7}$$

The target complex conjugate pole pair is -1 ± 2j specified in [15] where as average distance error e is calculated using distance formula for two numbers. For evolved eigen values -0.78 ± 1.063j the maximum average distance error turns out to be 0.961.

Using values of settling time and damping ratio from equations 4 and 7 respectively we have natural frequency of the system ω_n equal to 1.319 rad/s and damped natural frequency of the system thus becomes ω_d = 1.064 rad/s [1]. Value of time period τ is calculated as 5.904 s/rad. Value of percent overshoot %OS = 10% is determined using equation 8 same as observed from the output curve in Figure 4.

$$\%OS = e^{-\xi\pi/\sqrt{1-\xi^2}} \tag{8}$$

Generic second order system transfer function *G(s)* is given by the following equation.

$$G(s) = \frac{K\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \tag{9}$$

After simplification the system transfer function *G(s)* is given by equation 10. Here value of K or steady state gain is taken as unity.

$$G(s) = \frac{1.739}{s^2 + 1.56s + 1.739} \tag{10}$$

From equation 10 the characteristic equation for this particular system thus becomes *F(s)* = s^2 + 1.56s + 1.739. Using quadratic formula and solving for roots of the system equation or poles of the physical system we have $s_{1,2}$ = -0.78 ± 1.063j. The rise time, settling time and damping ratio are typical of this type of systems. Value of attenuation σ is determined as 0.77 rad/s, β is 54° and maximum overshoot M_p is 10.3%.

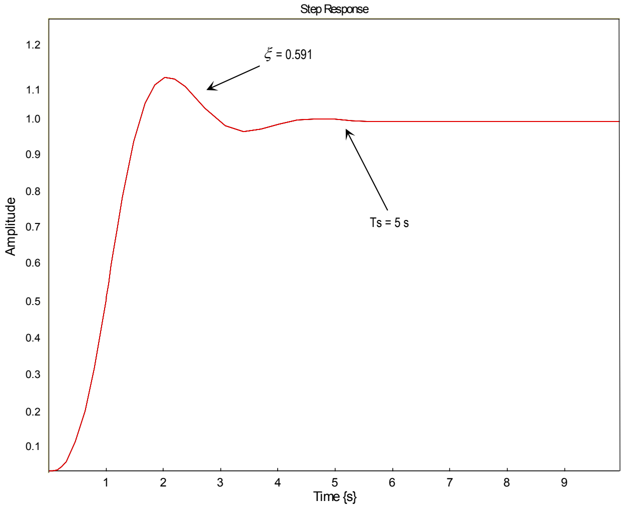

Figure 4. System response to a unit step input.

4. SUMMARY/CONCLUSION

The methodology followed has been proposed for unified and automated design of mechatronic or multi domain dynamic systems using Bond-Graphs for system representation and genetic programming for exploring the design space in an open ended manner. This research paper has been a product of an indigenous attempt to implement the design methodology for a postgraduate level research project carried out in two stages. At first stage the objective was to repeat and/or develop a simple experiment based on the said methodology and implementation scheme to achieve comparable results [7]. As the C language code was developed without referring to any source so certain limitations were unavoidable resulting in final eigen values falling slightly short of the target. However the implementation and results prove that the methodology is valid and thus verified. The physical system in Figure 3 is an intuitive rotary mechanical interpretation of the Bond-Graph model. For second stage of the project dynamic analysis of the evolved system has been carried out to illustrate that a stable physical system can be realized from the open ended synthesis paradigm considered so inherent in all applications of automated design concept using genetic programming. The robustness of the unified/automated design approach lies in compactness of the genetic programming code and fitness evaluation of the evolved designs. The interpretation of the results will be simplified in further research efforts. The complexity of the implementation especially code development and final elucidation is one of the reasons that this research area remains relatively less explored. So far results of only one research group have been published [15], [16]. The problem implemented in this research paper though simple gives perhaps the first independent verification of the design methodology identified by this particular research group. It is intended that this methodology will be followed for extending the automated design concept to more sophisticated mechatronic systems like kinematic sections of humanoid robots, end effectors and even synthesis of a simple two legged walking robot. An observation pertaining to Bond-Graphs based representation is the limitation imposed due to lack of two port elements transformer *TF* and gyrator *GY* on the automated synthesis process which tends to restrict the evolutionary search and synthesis to one particular energy domain at any time without scaling. Direct transition from one energy domain to the other can be made if a genetic programming function for gyrator element is available thus extending the range of the design approach to nearly complete multi energy domain systems.

5. ACKNOWLEDGMENTS

I thank for the continued encouragement of Dr. Muhammad Afzaal Malik and generosity extended by Dr. Muddassar Farooq regarding a previous research publication. This project was carried out at College of Electrical and Mechanical Engineering, National University of Sciences and Technology, Rawalpindi, Pakistan as part of and extension to a postgraduate research thesis. The help of the institution is appreciated without which this work would not have been possible.

6. REFERENCES

[1] Burns, R. S. *Advanced Control Engineering*, Butterworth-Heinemann, 2001.

[2] Fan, Z., Seo, K., Rosenberg, R. C., Hu, J., and Goodman, E. D. Computational Synthesis of Multi Domain Systems, In *Proceedings of AAAI Spring Symposium on Computational Synthesis*, Stanford, 2003, 59-66.

[3] Hirsh, H., Banzhaf, W., Koza, J. R., Ryan, C., Spector, L., and Jacob, C. Genetic Programming. *IEEE Intelligent Systems*, 15(3), (2000), 74-84.

[4] Hu, J. *Sustainable Evolutionary Algorithms and Scalable Evolutionary Synthesis of Dynamic Systems*. PhD Dissertation, Department of Computer Science and Engineering, Michigan State University, East Lansing, MI, USA, 2004, 69-80.

[5] Karnopp, D. C. and Margolis, D. L. The Language of Interaction. *ME Magazine*, American Society of Mechanical Engineers, January 2001, 1-4.

[6] Karnopp, D. C., Margolis, D. L., and Rosenberg, R. C. *System Dynamics: Modeling and Simulation of Mechatronic Systems*, John Wiley, 2000.

[7] Kayani, S. A. and Malik, M. A. Combining Bond-Graphs with Genetic Programming for Unified/Automated Design of Mechatronic or Multi Domain Dynamic Systems, In *Proceedings of Ninth Genetic and Evolutionary Computation Conference*, London, 2007, 2515-2518.

[8] Kayani, S. A. On Automated Design of Mechatronic Systems through Bond-Graphs and Genetic Programming. *IEEE Multi Disciplinary Engineering Education Magazine*, Vol. 2, No. 4, (December 2007), 15-17.

[9] Koza, J. R., Bennett, F. H., Lohn, J., Dunlap, F., Keane, M. A., and Andre, D. Automated Synthesis of Computational Circuits using Genetic Programming. In *Proceedings of IEEE International Conference on Evolutionary Computation*, Indianapolis, 1997, 447-452.

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

[11] Koza, J. R., Keane, M. A., and Streeter, M. J. What's AI Done for Me Lately? - Genetic Programming's Human Competitive Results. *IEEE Intelligent Systems*, 18(3), (2003), 25-31.

[12] Novales, A. C. *Genetic Programming Studio 1.0 User Manual*. University of Cordoba, Spain, 1998.

[13] Ogata, K. *Modern Control Engineering*, Prentice Hall, 1997.

[14] Palm, W. J. *Control Systems Engineering*, John Wiley, 1986.

[15] Seo, K., Hu, J., Fan, Z., Goodman, E. D., and Rosenberg, R. C. Automated Design Approaches for Multi Domain Dynamic Systems using Bond-Graphs and Genetic Programming. *The International Journal of Computers, Systems and Signals*, Vol. 3, No. 1, (2002), 55-70.

[16] Wang, J., Fan, Z., Terpenny, J. P., and Goodman, E. D. Knowledge Interaction with Genetic Programming in Mechatronic Systems Design using Bond-Graphs. *IEEE Transactions on Systems, Man and Cybernetics: Applications and Reviews*, Vol. 35, No. 2, (May 2005), 172-182.

[17] Zongker, D. and Punch, W. *lil-gp 1.01 User Manual.* Michigan State University, East Lansing, MI, USA, 1996.