Open-ended Robust Design of Analog Filters Using Genetic Programming

Jianjun Hu
Department of Computer Science
Purdue University
West Lafayette, 47906, IN
hujianju@purdue.edu

Xiwei Zhong
Huzhang University of Science & Technology
Wuhan, Hubei, 430074
China
xiweizhong@hotmail.com

Erik D. Goodman
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, MI, 48824
goodman@egr.msu.edu

ABSTRACT

Most existing research on robust design using evolutionary algorithms (EA) follows the paradigm of traditional robust design, in which parameters of a design solution are tuned to improve the robustness of the system. However, the topological structure of a system may set a limit on the possible robustness achievable through parameter tuning. This paper proposes a new robust design paradigm that exploits the open-ended topological synthesis capability of genetic programming to evolve more robust systems. As a case study, a methodology for automated synthesis of dynamic systems, based on genetic programming and bond graph modeling (GPBG), is applied to evolve robust low-pass and high-pass analog filters. Compared with a traditional robust design approach based on a state-of-the-art real-parameter genetic algorithm (GA), it is shown that open-ended topology search by genetic programming with a fitness criterion rewarding robustness can evolve more robust systems with respect to parameter perturbations than what was achieved through parameter tuning alone, for our test problems.

Categories and Subject Descriptors: G.1.6 [Global Optimization]; I.2.2 [Automated Programming] ; I.2.1 [Application] ;

General Terms: Design, algorithms.

Keywords: Genetic programming, robust design, analog filter synthesis, bond graphs, automated design.

1. INTRODUCTION

Topologically open-ended computational synthesis by genetic programming (GP) has been used as an effective approach for engineering design innovation, with many success stories in a variety of domains including analog circuits, digital circuits, molecular design, mechatronic systems, etc. [15][18]. These works employ GP as an open-ended search method for functional design innovation – achieving given behavior without pre-specifying the design topology – and has achieved considerable success. However, in practical engineering system design, there is another criterion in addition to functional specifications that should be considered during the design process. Robustness, as the ability of a system to maintain function even with changes in internal structure (including variations of parameters from nominal values) or external environment [5],[10], is also critical to engineering design decisions. Engineering design systems, in reality, do not normally take into account all the types of uncertainties or variations to which the engineered artifacts are subject, such as manufacturing variation, degradation or non-uniformity of material properties, environmental changes, and changing operating conditions. There are two types of robustness of dynamic systems that we are interested in. One is the robustness of systems with respect to perturbation of the parameters of the system. This is the most commonly investigated type of robustness in the traditional robust design community and also in evolutionary robust design. Another type of system robustness is with respect to topological perturbation – for example, accidental removal or failure of components. Reliable systems, having the least sensitivity of performance to variations in the system components or environmental conditions, are very desirable. However, there are relatively few studies that explore how GP-based open-ended topology search may contribute to design of robust systems such that they can withstand internal or external perturbations. In traditional robust design, optimizing robustness is usually regarded as a step in the detailed design stage, in which the parameters of a system with a given functional structure are tuned to achieve better robustness.

We are interested in applying GP-based open-ended design synthesis to robust engineering design. Specifically, we examine whether topological innovation capability of GP can facilitate design of more robust dynamic systems with respect to parameter variations or uncertainty of the design variables. The robustness with respect to structural faults of the system is also being investigated, but will be reported elsewhere. In this paper, a set of experiments is conducted to test the following hypotheses about robust design using genetic programming: 1) that dynamic systems with high performance evolved by GP without considering a robustness criterion during the evolutionary process may have unacceptably low robustness with respect to parameter perturbation, 2) that the robustness of a system may be constrained by its topological/functional structure, and the amount of robustness improvement available through pa-

GECCO'05, June 25–29, 2005, Washington, DC, USA.

rameter tuning is limited as well, and 3) that topologically open-ended synthesis by GP may allow evolution of more robust solutions than the traditional robust design approaches with parameter tuning.

To examine the role of topology search in designing robust systems, two analog filter design problems, including low-pass and high-pass filters, are to be synthesized using genetic programming. For each synthesis problem, three types of experiments are conducted: a) evolutionary synthesis using GP without considering a robustness criterion, b) improving robustness of these evolved filters by tuning their parameters using a genetic algorithm (GA), and c) evolving robust filters (topological structure and parameters) using GP with a robustness criterion in the fitness function. These filter design problems are selected as they are perhaps the most popular problems in evolutionary synthesis research by either GA or GP [14] [16].

The rest of the paper is organized as follows. Section 2 presents an abbreviated survey of applications of evolutionary algorithms in robust design. Section 3 introduces the GPBG methodology, which exploits Genetic Programming and Bond Graphs for automated synthesis of dynamic systems. This section introduces some new features that improve standard developmental GP for bond graph synthesis. Section 4 discusses two approaches to evolving robust dynamic systems – the parameter search approach (by genetic algorithm) and the simultaneous topology and parameter search approach by genetic programming. Section 5 compares experimental results of these approaches. Finally, the conclusions and future research are highlighted in Section 6.

2. RELATED WORK

Robust design as originally proposed by Taguchi [19] has been intensively investigated in the engineering design community since the 1980s and remains an important topic. In traditional robust design, a designer seeks to determine the control parameter settings that produce desirable values of the mean (nominal) performance, while at the same time minimizing the variance of the performance [19]. However, most of these robust design studies assume that there already exists a design solution for a system and the task of robust design is to determine its robust operating parameters with respect to various kinds of variations. The relationship of topological or functional structure of a system to its robustness is often not treated. Especially, how robustness criteria should be incorporated into conceptual functional design stage is not addressed.

Application of EAs to traditional parametric robust design has been attracting increasing attention in the past decade [20] Wiesmann:1998 [8] [11]. Tsutsui et al. [20] proposed a GA-based Robust Solution Searching Scheme (RS^3) to evolve robust solutions. This approach works by adding perturbation noise to the design variables before fitness evaluation. In Wiesmann et al.'s approach [21], however, each individual is simulated t times to estimate its expected loss function (fitness). Their experiments showed that the evolved designs were substantially more robust to parameter variations than the reference design, but usually at the cost of reduced performance in undisturbed situations. This observation motivated the later work of using an evolutionary multi-objective approach to figure out the trade-off map between robustness and optimal functional performance [8] [17] [11]. Forouraghi[8] introduced an interval computation method to avoid artificial insertion of Gaussian noise to parameter variables in order to build tolerance against internal or external perturbations. Ray [17] expressed the robust design problems as a three-criterion multi-objective problem, simultaneously optimizing an individual's performance without perturbation, the mean performance of its neighbors resulting from perturbations, and the standard deviation of its neighbors' performances. Jin et al. [11] proposed two methods for estimating the robustness measures of an individual – by exploiting its neighbor individuals in the current population or by using all evaluated individuals. Jin's robustness estimation approach can greatly reduce the number of function evaluations, when it is applicable. However, it is difficult to apply this method for evolving robust designs with variable structures as in topologically open-ended automated synthesis using GP because of the difficulty to define a neighborhood for a given individual in the structural space.

We chose analog filter synthesis problems as our benchmarks since they are the most widely used test problems in electric circuit optimization using GA or GP [14] [16] [7]. The pioneering work of Koza in automated analog circuit synthesis, including low-pass, high-pass, and asymmetric band-pass filters, is described in [14] [13]. Lohn and Colombano [16] proposed a linear representation approach to evolve analog circuits in which several low-pass filters were used as test problems. However, they did not specifically work on evolving robust circuits. In our previous work [7], we applied GP to lowpass filter design problems using bond graphs as the modeling and simulation tool.

A lot of work has been done in both evolutionary robust design and analog circuit synthesis. However, there are few studies that specifically address how GP-based topologically open-ended synthesis may provide a new way for open-ended robust design. This may enable us to move robust design forward to the conceptual/functional design stage and thus achieve design for robustness at the very beginning, which will augment the current practice of design for robustness in parametric design.

3. ANALOG FILTER SYNTHESIS USING BOND GRAPHS AND GP

In this section, we present an improved methodology for open-ended computational synthesis of multi-domain dynamic systems based on bond graphs [12] and GP–the GPBG approach = Genetic Programming+Bond Graphs.

3.1 Bond Graphs

The bond graph is a multi-domain modeling tool for analysis and design of dynamic systems, especially hybrid multi-domain systems, including mechanical, electrical, pneumatic, hydraulic, etc., components. Details of notation and methods of system analysis related to the bond graph representation can be found in [12]. Fig. 1 illustrates a small bond graph that represents the accompanying electrical system. Fig. 2 shows the complex bond graph model of a low-pass filter. A typical simple bond graph model is composed of inductors (I), resistors (R), capacitors (C), transformers (TF), gyrators (GY), 0-Junctions (J0), 1-junctions (J1), sources of effort (SE), and sources of flow (SF). In this paper, we are only concerned with linear dynamic systems and did not include transformers and gyrators as components.

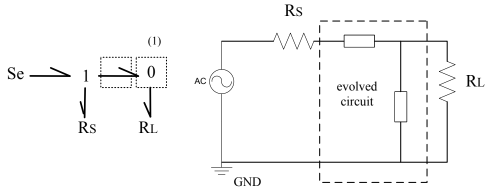

Figure 1: A bond graph and its equivalent circuit. The dotted boxes in the left bond graph indicate modifiable sites at which further topological manipulations can be applied (to be explained in next section)

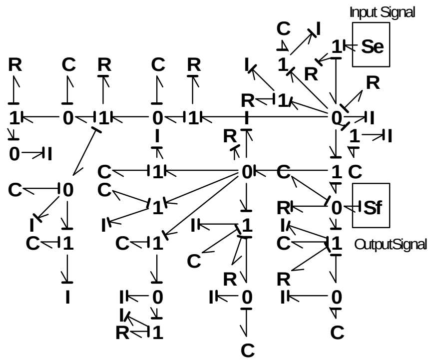

Figure 2: Bond graph structure of low-pass filter evolved in 500,000 function evaluations. Filter has 39 components beyond embryo. (Component sizes omitted for simplicity.)

3.2 Evolving Analog Filters using Bond Graphs and GP: the GPBG framework

Automated synthesis of bond graphs involves two basic searches: the search for a good topology and the search for good parameters for each topology, in order to be able to evaluate its performance. Based on the pioneering work of Koza [13] on automated synthesis of electronic circuits, we created a developmental GP system for synthesizing mechatronic systems represented as bond graphs[18]. This GPBG framework enables us to do simultaneous topology and parameter search.

The GPBG framework includes the following major components: 1) an embryo bond graph with modifiable sites at which further topological operations can be applied to grow the embryo into a functional system, 2) a GP function set, composed of a set of topology manipulation and other primitive instructions which will be assembled into a GP tree by the evolutionary process (execution of this GP program leads to topological and parametric manipulation of the developing embryo bond graph), and 3) a fitness function to evaluate the performance of candidate solutions.

In this paper, we have improved the basic function set in [7] and developed a hybrid function set to reduce redundancy while retaining flexibile topological exploration:

```
F={ Insert_J0E, Insert_J1E, Add_C/I/R, EndNode,
          EndBond, ERC}
```

where the Insert_J0E, Insert_J1E (Fig. 3) functions insert a new 0/1-junction into a bond while attaching at least one and at most three elements (from among C/I/R). EndNode and EndBond terminate the development (further topology manipulation) at junction modifiable sites and bond modifiable sites, respectively; ERC represents a real number that can be changed by Gaussian mutation. In addition, the number and type of elements attached to such junctions are controlled by three bits. A flag mutation operator is used to evolve these flag bits, each representing the presence or absence of corresponding C/I/R components. This hybrid approach does not create the many bare (and unnecessary) junctions generated by the basic approach. At the same time, Add_C/I/R still provide the flexibility needed for broad topology search. For any of the three C/I/R components attached to each junction, there is a corresponding parameter to represent the component's value, which is evolved by a Gaussian mutation operator in the modified GP system used here. Fig. 4 shows a GP tree that develops an embryo bond graph into a complete bond graph solution. The comparison experiments of [9] showed that this function set was more effective on both an eigenvalue and an analog filter test problem, so the new set was used in this paper.

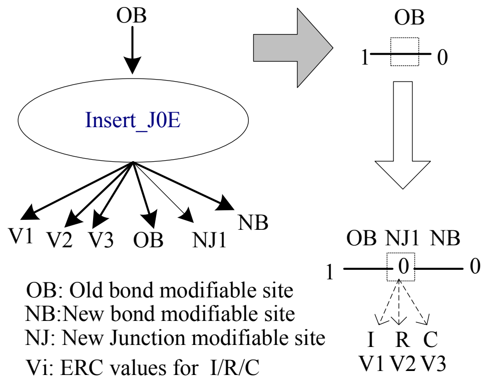

Figure 3: The Insert_J0E function inserts a new junction into a bond along with a certain number of attached components

As a case study, we are interested in evolving two types of analog filters including low-pass and high-pass filters (Fig. 5). In these GPBG based filter design problems [7], a bond-graph-represented analog filter composed of capacitors, resistors, and inductors is to be evolved such that the magnitude of its frequency response approximates a specified filter frequency response specifications. An embryo bond graph and its equivalent circuit are illustrated in Fig. 1. This embryo bond graph is used in all three filter design problems. Note that the 0-junction is the modifiable site, where further topological developments can proceed as instructed by a GP program tree. The voltage of this 0-junction is the output signal.

Rather than using a sophisticated SPICE simulator as is often done in analog filter synthesis [14] [1], calculation of

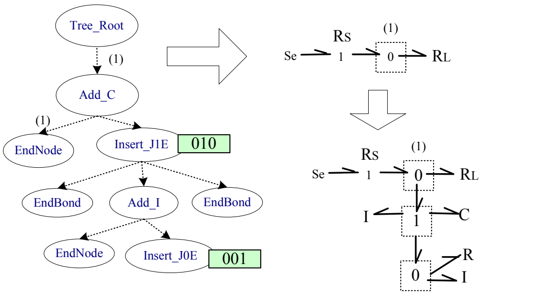

Figure 4: Sample GP tree evolved by applying topology operators to embryo, generating a bond graph after depth-first execution (numeric ERC nodes omitted). Flag bits 010 and 001 show presence or absence of attached C/I/R components.

frequency response from a bond graph was done by automatically formulating the state equations (yielding A, B, C, and D matrices), then using MATLAB 3.0-derived C++ code to simulate behavior.

Detailed specifications of the filter synthesis problems are:

- The low-pass filter synthesis problem is extracted from [14], in which the frequency response performance of a candidate filter is defined as the weighted sum of deviations from ideal frequency response magnitude over 101 points:

$$F_{koza}(t) = \sum_{i=0}^{100} [W(d(f_i), f_i) * d(f_i)] \qquad (1)$$

  Where f_i is the sampling frequency. $d(x)$ is the absolute deviation of candidate frequency response from target response at frequency x. W(x,y) is the weight function. The sampling points range from 1Hz to 100K Hz, logarithmically distributed. If the deviation from ideal magnitude is less than 0.03V, the weight is 1. If the deviation is more than 0.03V, the weight is 10. The pass band is [1,1K] Hz; the stop band is [2K,10K] Hz. A "don't" care band between 1K Hz and 2K Hz neglects any deviation from the target response.

- The high-pass problem is similar, except for the complementary definitions of the pass and stop bands. The passband is now defined as [2K,10K]Hz, while the stopband is [1,1K]Hz. A "don't care" band from 1KHz to 2KHz neglects any deviation from the target response.

To evolve an analog filter without considering robustness, the final fitness of a candidate solution is defined as follows:

First, calculate the raw fitness of a candidate solution defined as the average absolute deviation between the frequency response magnitude of the candidate solution and the target frequency response over all 101 sampling frequencies:

$$f_{raw} = \frac{1}{100} \cdot F_{koza}(t) = \frac{1}{100} \cdot \sum_{i=0}^{100} [W(d(f_i), f_i) * d(f_i)] \quad (2)$$

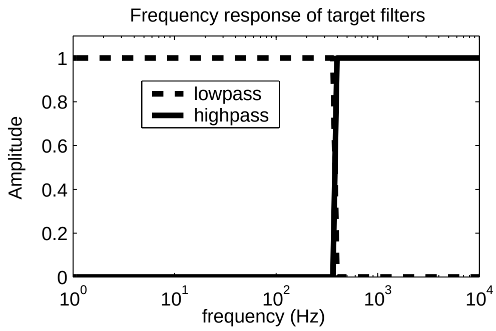

Figure 5: Specification of analog low-pass and high-pass analog filter synthesis problem.

Note that this f_{raw} definition differs from Koza's raw fitness definition in Equ.1 by a multiplier equal to the number of sampling frequencies. We use the average deviation rather than sum of deviations to remove the influence of the number of sampling points.

Then calculate the final fitness as:

$$f_{norm} = \frac{NORM}{NORM + f_{raw}} \qquad (3)$$

where NORM adjusts fnorm into the range [0,1]. This transforms the problem of minimizing deviation from target frequency response into a maximization problem appropriate for our GP system. Since tournament selection is used, NORM can be an arbitrary positive number (here set to 10, yielding fitness ranges around [0, 1]).

3.3 Modified Developmental GP

Compared to the GP systems used in [13], we use a standard strongly-typed multi-population generational GP with the following modifications: a flag bit mutation operator is introduced to evolve the configuration of C/I/R elements attached to a junction; a subtree-swapping operator is used to exchange non-overlapping subtrees of the same individual (GP tree); an ERC mutation operator is developed to evolve the parameter values for all C/I/R components; elitism is used throughout the evolution process. The motivation of these modifications is to allow more flexible topology modification and better parameter search.

4. EVOLVING ROBUST ANALOG FILTERS USING BOND GRAPHS AND EA

There are two interesting types of system robustness. One is robustness with respect to variation of parameter values of the components; the other is robustness with respect to failure of components. In this paper, only the first type is examined, to show how topology innovation can improve upon traditional robust design methodology. In mechanical systems where bond graphs are widely used, actual component dimensions are often constantly changing due to friction, wear, and damage, and thus robustness with respect to parameter variation is highly desirable.

We used three methods to evolve robust or non-robust lowpass and highpass filters. The first is a standard GP approach without considering robustness requirements. The second is a hybrid GP/GA robust design method (GP-GARMS).

This approach first uses a standard GP to evolve a high-performance filter without incorporating any robustness criterion in the fitness function. And then the state-of-art G3PCX-GA is used to improve the robustness of the GP solution using the multi-simulation method to evaluate the robustness performance. The third is a standard GP with multi-simulation (GPRMS), which uses multiple simulations to estimate the robustness fitness of a candidate solution.

4.1 Evolving Robust Filters Against Parameter Variation: the Unified Approach

The typical approach for evolving robust designs [3] is to use multiple Monte Carlo samplings with different environmental or system configurations (e.g., perturbation of parameter values of the system) to calculate a worst-case or an average fitness for a given candidate solution as shown in Equ. (3). This robust-by-multiple-simulation (RMS) method is used in [21]. Another method is to simply add a perturbation to the design variable before evaluation. This perturbation, however, is not incorporated into the genome, making it different from normal parameter mutation operator or Larmarckian style evolution algorithms. This robust-by-perturbed-evaluation (RPE) method is used in [20] and is suggested to be more efficient by Jin et al. [11]. Both methods are tested in this work. For RPE method, no special fitness definition is needed. For the RMS multi-simulation method, our raw fitness for a design solution with robustness criterion is defined as follows:

$$f_{robustraw} = \sum_{k=1}^{SPI} f_{raw}^{k} \tag{4}$$

where SPI is the number of Monte Carlo sampling evaluations for each individual, f_{raw}^{k} is the raw fitness of the kth sampled evaluation with a different Monte Carlo perturbation of the parameters as defined in 2. With this raw robustness fitness, we then calculate the final fitness according to Equ. 3.

The perturbation of the component values during evolution in the experiments reported below is implemented by adding to each component parameter Gaussian noise $N(\mu, \sigma)$ with mean μ of 0 and standard deviation σ set at 10% of the parameter value. This perturbation model is widely used by previous researchers [20] [21] and may not be appropriate for all manufacturing processes. However, it is good enough for our purpose as an approximation to the real component value degradation model in some situations. If the parameter value is ever 0, σ is set to 1.

In the evolution stage of RMS method, the number of Monte Carlo samplings for fitness evaluation of each individual with respect to parameter perturbation is set as SPI =10. After the robust solutions are evolved, their robustness with respect to parameter perturbation is evaluated against a series of perturbation magnitudes: Gaussian noise $N(\mu, \sigma)$ with mean μ set at 0 and standard deviation σ set at 10% to 50% of parameter values in steps of 10%, each tested with 10000 samplings with different configurations of the component parameter perturbations.

4.2 Evolving Robust Analog Filters Using GA: the Traditional Robust Design

Evolutionary algorithms have been increasingly applied to evolve robust designs [2],[21] or for optimization in noisy environments [3],[4]. Most such research follows the practice of traditional robust design: given a system with a specific functional structure, tune its parameters using evolutionary algorithms to improve robustness.

We shall contrast the traditional approach described in this section with the new approach. We shall first evolve a high-performance analog filter with the improved GPBG approach as described in Section 3.2. No requirement for robustness is enforced during this evolution. Then we shall apply to the result a state-of-art real parameter genetic algorithm–the G3PCX-GA proposed by Deb [6]–to tune the parameters of this filter to improve its robustness with respect to parameter perturbation while keeping its functional structure unchanged.

In the minimization G3PCX-GA, we used the robust raw fitness defined in Equ. 4 as the final fitness of an individual. We believe this fitness measure is better than the average of normalized final fitness of each sampling evaluation for its lower distortion of the optimization objective values.

4.3 Evolving Robust Analog Filters Using GP: A New Robust Design Paradigm

This new approach aims at exploiting the topology search capability of GP to evolve more robust designs. The configuration of this approach is the same with standard GP-based synthesis except that the robustness criteria is incorporated in the fitness function. The final fitness of an individual, calculated from the sampling fitnesses, is the same as defined in (2), where f_i^k is defined as $Fitness_{norm}$ in (3).

5. EXPERIMENTS AND RESULTS

For all experiments below, a fixed number of function evaluations is allocated to ensure fairness of comparison. For the lowpass and highpass filter design problem, the computation budget is 1,000,000 function evaluations. Note that for methods that use multiple simulation to estimate the robustness fitness, each simulation is counted as one function evaluation. In addition, for the hybrid GP-GARMS method, we allocate 500,000 for GP evolution and the remaining 500,000 for GA evolution for robustness.

All experiments described below used the same embryo bond graph shown in Fig. 1. The component values of source resistor R_s and load resistor R_{load} are both 1 Ω for lowpass and highpass filter synthesis. Our GPBG based system is implemented with C++. The GP code is based on modified Open Beagle. The bond graph simulator was developed in our lab. All experiments were run on a single Linux machine with a 3.0GHz CPU and 1GB memory. On average, for an experiment run with 1,000,000 fitness evaluations, it took about 10-20 hours depending on the complexity of the GP trees. To make our results to be practical, we intentionally used a single set of parameters as much as possible to run all experiments with little tuning effort.

To assess the statistical significance of the performance differences, for each target filter type and each synthesis method, 10 runs were conducted. This size of experiments is determined by the computing resources available. However, since the results are quite stable across runs, it is sufficient for the purposes of this paper. In addition, due to the page limitation of this paper, only intuitive results with brief statistical tests are reported.

5.1 Evolving Analog Filters using GP Without Considering Robustness

In this experiment, ten analog lowpass and highpass filters were evolved using standard GP without incorporating a robustness criterion in the fitness function (3). The following common running parameters were used throughout all GP experiments in this paper Table 1:

We select the evolved filter with the highest performance to test its noise tolerance over the degradation or variation of the component parameters with different perturbation magnitudes. As described above, the evaluation of robustness with respect to parameter perturbation is conducted by running 10000 simulations of the configurations of the Gaussian parameter perturbations.

Fig. 2 and Fig. 6 show the topologies of the evolved lowpass and highpass filters with highest performance out of ten runs. The evolved best lowpass and highpass filters have 39 and 27 components, respectively. The lowpass and highpass filters approximate the ideal frequency response closely, with the sum of deviation over 101 points being only 6.43 and 0.32, respectively.

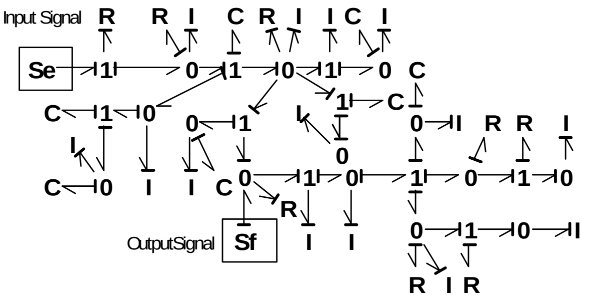

Figure 6: Topology of best highpass analog filters evolved with standard GP with 500,000 function evaluations without considering robustness requirement.This filter has 27 C/I/R components beyond original embryo. The best evolved lowpass filter is shown in Fig. 2. This topology is generated by a simplification procedure which removes redundancy in the original evolved bond graphs while maintaining their functional behaviors.

5.2 Evolving Robust Analog Filters Using GA: the Classical Robust Design

In this experiment, the G3PCX-GA is used to improve the robustness of the best analog filter evolved in the previous section through parameter tuning while keeping functional structure unchanged. As we can see from Fig. 2 and Fig. 6, these two filter models are very complex, with 39 and 27 parameters to search. As the objective function of this optimization is highly multi-modal, this is a hard optimization even for G3PCX-GA, as the experiment demonstrates. The running parameters for this experiment are summarized in Table II.

Fig. 7(c) and Fig. 8(c) show the twenty frequency response curves of these robust lowpass and highpass filters with parameter perturbations of 30% of nominal values. Compared with the result in Fig. 7(b) and Fig. 8(b) without considering robustness, the G3PCX GA indeed improves the robustness. However, one needs to be cautious when interpreting these frequency response figures. As specified in our synthesis problems, we have a shallow "don't care" region for both lowpass and highpass target frequency responses. The robust filters evolved by G3PCX have better performance in the two-end regions, while they have large variation in the "don't care" region.

5.3 Evolving Robust Filters Using GP: Open-Ended Topology Search for Robust Design

In this section, we try to evolve robust analog filters that have higher tolerance of the variation of component values and have graceful performance degradation. The configuration of this experiment is the same as those used in Section 5.1.

The topology of the evolved robust lowpass and highpass filters are shown in Fig. 7(a) and Fig. 8(a). It is very interesting to compare the complexity of these two filters to those evolved with standard GP without considering robustness (Fig. 2 and Fig. 6). The robustness requirement drives the GP to evolve much simpler structures since large structures expose more components to perturbation noise. Of course, this depends on the perturbation model. In our model, we applied the perturbation to ALL components, so large filters with more components tend to suffer from more perturbation.

We can also compare the frequency responses of the robust filters with that of filters evolved by standard GP and GA with robustness. It appears that GP with robustness beats the other two results by allowing more variation in the "don't care" region while keeping tight control in the two boundary regions where stringent functional requirements are imposed.

5.4 Statistical Comparison of Three Methods

For the highpass filter problem, we did a t-test to the compare the robustness of the evolved solutions by GPGARMS and standard GP in terms of fitness variation at 0.2 perturbation level. A significance level of $P =< 0.001$ is achieved strongly indicating GPGARMS improved the robustness of the evolved filters by standard GP. However, we found that this improvement is at the cost of degraded performance. A t-test was also applied to compare GPGARMS and GPRMS. The 95 percent confidence interval for difference of means of robustness fitness is -51.617 to -39.841, showing that GPGARMS degraded robustness. The difference in the mean values of the two groups is greater than would be expected by chance ($P =< 0.001$).

6. CONCLUSIONS AND FUTURE WORK

This paper applies GP and bond-graph-based modeling – the GPBG approach – to topologically open-ended synthesis of robust dynamic systems. It is shown that the traditional approach of robust design, in which the functional conceptual design is conducted without considering a robustness requirement, may put severe limits on the possible robustness achievable through parameter-tuning-based robust design during the detailed design stage. It thus proposes that robust design in engineering should start from the conceptual stage, and that the open-ended topology search capability of GP can be exploited for this purpose. We find that our GP system enables us to find more robust analog filters with respect to the variations in their parameters compared

Table 1: Experimental parameters for analog filter synthesis without robustness criterion

Total population size: 2000(400/400/400/400/400)	Number of subpopulations: 5
migration interval: 5 generation	Migration size: 30 individuals
Max tree depth: 8	Crossover probability: 0.7
InitTreeDepth: 3-5	Standard mutation probability: 0.1
flag bit mutation rate: 0.1	swapping-tree mutation rate: 0.1
Tournament size: 7	Parametric mutation probability: 0.5
Max evaluations: 1,000,000	Flag mutation probability: 0.3
Pool size of elite individuals: 20	Elite pool update frequency: 5 generations

Table 2: Experimental parameters for robust design by G3PCX-GA

Total population size: 200	Max evaluations: 500,000
Number of parents in crossover: 3	family size: 2
$sigma_z eta$: 0.1	$sigma_e ta$: 0.1
SPI: 10	Perturbation noise percentage: 20%

to existing parameter-tuning-type evolutionary algorithms for robust design of fixed functional structures.

Evolving robustness is a rich research theme and there are several interesting topics to be further investigated. For example, another dimension of system robustness which is not discussed in this paper is the robustness with respect to topology perturbation or component failures, which may be important in many environments, especially on space missions. Our ongoing work shows that selection pressures for robustness with respect to parameter perturbation versus with respect to component faults lead to different topological patterns. It would be interesting to investigate how simultaneous requirements for both types of robustness would affect topological structures. In this paper, only simple robustness estimation method based on multiple sampling is used. However, in the simultaneous topology and parameter search process, more effective approach can be devised to reduce the computational effort for estimating robustness.

Acknowledgment

We sincerely acknowledge the beneficial discussion and support from Dr. Daisuke Kihara of Purdue University.

7. REFERENCES

[1] S. Ando and H. Iba. Linear genome methodology for analog circuit design. Technical report, Information and Communication Department, School of Engineering, University of Tokyo, 2000.

[2] J. Branke. Creating robust solutions by means of an evolutionary algorithms. In A. Eiben, T.Back, M. Schoenauer, and H.-P. Schwefel, editors, *Proc. Parallel Problem Solving from Nature. number 1498 in LNCS. Springer*, 1998.

[3] J. Branke. *Evolutionary Optimization in Dynamic Environments.* Kluwer, 2001.

[4] J. Branke and H. Schmeck. Designing evolutionary algorithms for dynamic optimization problems. In S.Tsutsui and A. Ghosh, editors, *Theory and Application of Evolutionary Computation: Recent Trends*. Springer, 2002.

[5] J. M. Carlson and J. Doyle. Complexity and robustness. *Proceedings of National Academy of Science (PNAS)*, 99(1):2538–2545, 2002.

[6] K. Deb, A. Anand, and D. Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation*, 10(4):345–369, 2002.

[7] Z. Fan, J. Hu, K. Seo, E. D. Goodman, R. C. Rosenberg, and B. Zhang. Bond graph representation and GP for automated analog filter design. In E. D. Goodman, editor, *2001 Genetic and Evolutionary Computation Conference Late Breaking Papers*, pages 81–86, San Francisco, California, USA, 9-11 July 2001.

[8] B. Forouraghi. A genetic algorithm for multiobjective robust design. *Applied Intelligence*, 12:151– 161, 2000.

[9] J. Hu, E. Goodman, K. Seo, Z. Fan, and R. Rosenberg. The hierarchical fair competition (hfc) framework for sustainable evolutionary algorithms. *Evolutionary Computation*, 13(2), 2005.

[10] E. Jen. Definitions of robustness. santa fe institute robustness site, rs-2001-009. 2001.

[11] Y. Jin and B. Sendhoff. Trade-off between optimality and robustness: An evolutionary multi-objective approach. In C. F. et al., editor, *Proceeding of the Second Int. Conf. on Evolutionary Multi-criterion Optimization*, pages 237–251. Springer.

[12] D. Karnopp, D. L. Margolis, and R. C. Rosenberg. *System Dynamics: Modeling and Simulation of Mechatronic Systems.Third Edition.* John Wiley & Sons, Inc., New York, 2000.

[13] J. R. Koza, D. Andre, F. H. Bennett III, and M. Keane. *Genetic Programming 3: Darwinian Invention and Problem Solving.* Morgan Kaufman, Apr. 1999.

[14] J. R. Koza, F. H. Bennett III, D. Andre, M. A. Keane, and F. Dunlap. Automated synthesis of analog electrical circuits by means of genetic programming. *IEEE Transactions on Evolutionary Computation*, 1(2):109–128, July 1997.

[15] J. R. Koza, M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu, and G. Lanza. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence.* Kluwer Academic Publishers, 2003.

[16] J. Lohn and S. Colombano. A circuit representation technique for automated circuit design. *IEEE Transactions on Evolutionary Computation*, 3(3):205–219, 1999.

[17] T. Ray. Constrained robust optimal design using a multi-objective evolutionary algorithm. In *Proceeding of Congress on Evolutionary Computation*, pages 419–424. IEEE press, 2002.

[18] K. Seo, Z. Fan, J. Hu, E. D. Goodman, and R. C. Rosenberg. Toward an automated design method for multi-domain dynamic systems using bond graph and genetic programming. *Mechatronics*, 13(8-9):851–885, 2003.

[19] G. Taguchi. *Taguchi on Robust Technology Development: Bringing.* ASME, ASME, 1993.

[20] S. Tsutsui and A. Ghosh. Genetic algorithms with a robust solution searching scheme. *IEEE Trans. Evolutionary Computation*, 1(3):201–208, 1997.

[21] D. Wiesmann, U. Hammel, and T. Back. Robust design of multilayer optical coatings by means of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 2(4):162–167, 1998.

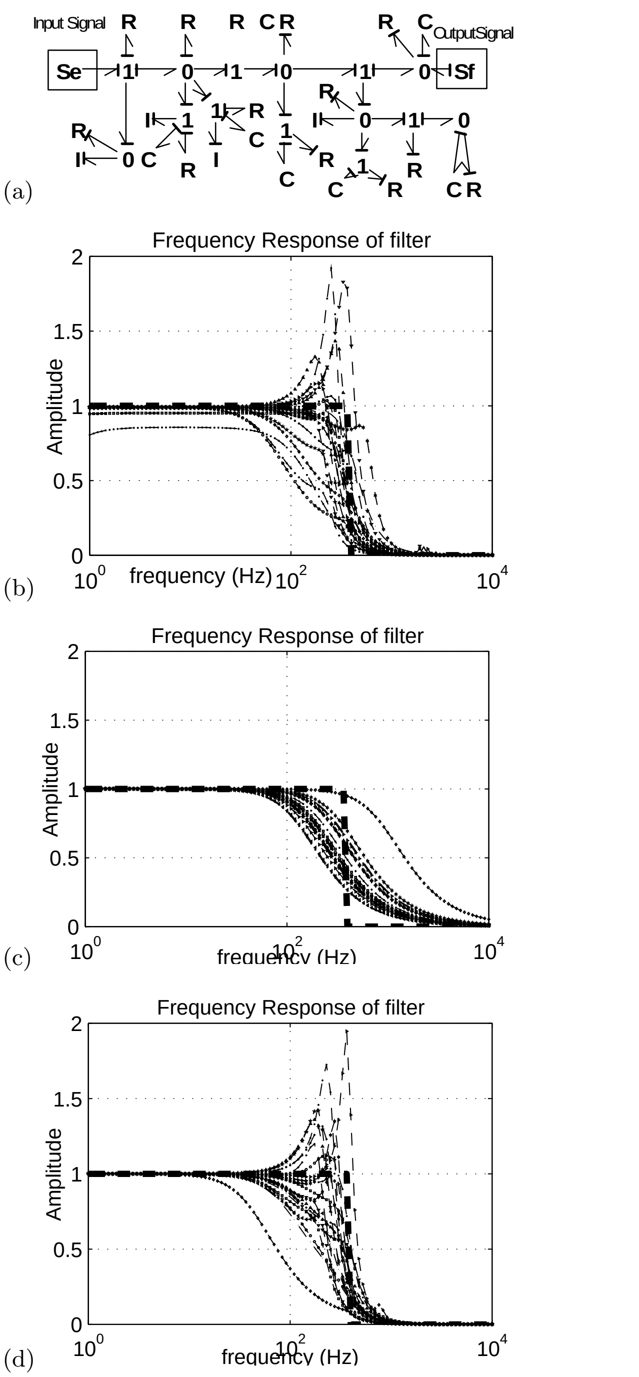

Figure 7: Structure of the robust lowpass filter evolved using GP with robustness requirements and its frequency responses in face of 30% Gaussian perturbation of their nominal parameters. (a) Robust lowpass filter evolved using GP with robustness (b) Frequency responses of the lowpass filter evolved using normal GP without robustness requirements (c) Frequency responses of the lowpass filter evolved first using normal GP without robustness requirements and then fine-tuned using GA with robustness requirements. (d) Frequency responses of the lowpass filter evolved using normal GP with robustness requirements

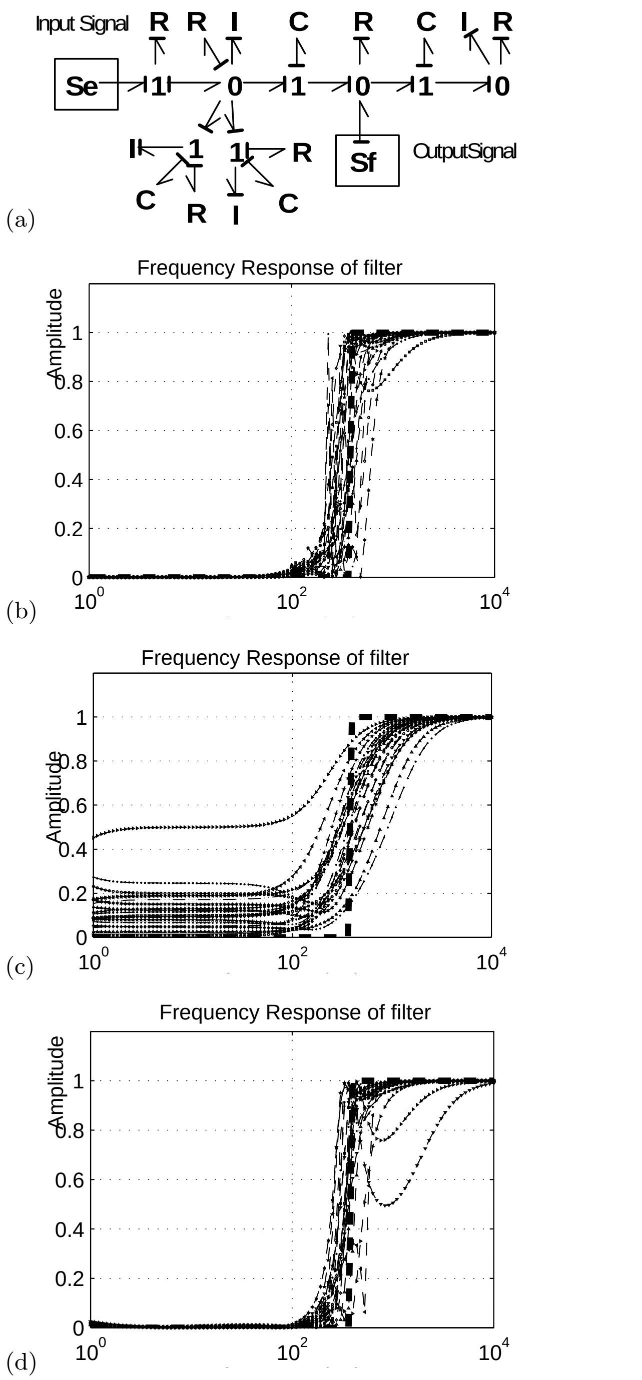

Figure 8: Structure of robust highpass filter evolved using GP with robustness requirements and its frequency responses in face of 30% Gaussian perturbation of their nominal parameters. (a) Robust highpass filter evolved using GP with robustness (b) Frequency responses of the highpass filter evolved using normal GP without robustness requirements (c) Frequency responses of the highpass filter evolved first using normal GP without robustness requirements and then fine-tuned using GA with robustness requirements. (d) Frequency responses of the highpass filter evolved using normal GP with robustness requirements