Chapter 11

ROBUST ENGINEERING DESIGN OF ELECTRONIC CIRCUITS WITH ACTIVE COMPONENTS USING GENETIC PROGRAMMING AND BOND GRAPHS

Xiangdong Peng[1], Erik D. Goodman[1] and Ronald C. Rosenberg[2]

[1]*Gentic Algorithms Research and Applications Group (GARAGe), Dept. of Electrical and Computer Engineering, Michigan State University* [2]*Dept. of Mechanical Engineering, Michigan State University*

Abstract Genetic programming has been used by Koza and many others to design electrical, mechanical, and mechatronic systems, including systems with both active and passive components. This work has often required large population sizes (on the order of ten thousand) and millions of design evaluations to allow evolution of both the topology and parameters of interesting systems. For several years, the authors have studied the evolution of multi-domain engineering systems represented as bond graphs, a form that provides a unified representation of mechanical, electrical, hydraulic, pneumatic, thermal, and other systems in a unified representation. Using this approach, called the Genetic Programming/Bond Graph (GPBG) approach, they have tried to evolve systems with perhaps tens of components, but looking at only 100,000 or fewer design candidates. The GPBG system uses much smaller population sizes, but seeks to maintain diverse search by using "sustained" evolutionary search processes such as the Hierarchical Fair Competition principle and its derivatives. It uses stochastic setting of parameter values (resistances, capacitances, etc.) as a means of evolving more robust designs. However, in past work, the GPBG system was able to model and simulate only passive components and simple (voltage or current, in the case of electrical systems) sources, which severely restricted the domain of problems it could address. Thus, this paper reports the first steps in enhancing the system to include active components. To date, only three models of a transistor and one model of an operational amplifier (op amp) are analyzed and implemented as two-port bond graph components. The analysis method and design strategy can be easily extended to other models or other active components or even multi-port components. This chapter describes design of an active analog low-pass filter with fifth-order Bessel characteristics. A passive filter with the same characteristics is also evolved with GPBG. Then the best designs emerging from each of these two procedures are compared. [The runs reported here are intended only to document that the analysis tools are working, and to begin study of the effects of stochasticity, but not to determine the power of the design procedure. The initial runs did not use HFC or structure fitness sharing, which will be included as soon as possible. Suitable problems will be tackled, and results with suitable numbers of replicates to allow drawing of statistically valid conclusions will be reported in this paper, to determine whether interesting circuits can be evolved more efficiently in this framework than using other GP approaches.]

Keywords: genetic programming, active component, transistor, bond graph, robust design strategy, Bessel analog filter design

1. Introduction

GP has been effectively applied to topologically open-ended computational synthesis (Koza et al., 2003). Though system performance is an important criterion, robustness, as the ability of a system to maintain its target performance even with changes in internal structure (including variations of parameters from their specified values) or external environment (Carlson and Doyle, 2002), is also critical to engineering design decisions. If the designed system is robust with respect to parameter values, it can probably still run well in a relatively harsh environment.

As there are many factors that affect system performance, it is difficult to take all system uncertainties or variability into consideration in robust engineering design. Two kinds of system robustness are often considered. One kind, widely investigated in robust engineering design (Du and Chen, 2000), is system robustness with respect to perturbation of component parameter values. Another kind is system robustness with respect to topology perturbation, such as component failure, short circuiting, etc. In this chapter, only robustness to parameter perturbation is considered.

This chapter considers evolution of a particular type of dynamic system, an analog low-pass filter with active components, as a design environment in which to do preliminary examination of some hypotheses about robust evolutionary design. The synthesis tool used throughout is called GPBG, a genetic programming (GP) system that uses trees to specify operations for construction of a bond graph (BG), which is a multi-energy-domain representation for dynamic systems. This GPBG system has earlier been used by the authors for automated design of a number of types of dynamic systems (Fan et al., 2001).

The chapter is organized as follows. Section 2 presents a short survey of robust design and introduction of evolutionary computation in this field. Section 3 discusses the GPBG methodology, which applies genetic programming and bond graphs for automated synthesis of dynamic systems. Bond graph modeling of common-emitter transistors and op amps is also discussed. Section 4 discusses topologically open-ended evolution and new active components and operators for robust design. Section 5 compares experimental results of these approaches. Conclusions and future research are discussed in Section 6.

2. Related Work

A method for robust design, called the Taguchi Method, pioneered by Dr. Genichi Taguchi, has greatly improved engineering productivity (Tay and Taguchi, 1993). After its introduction, it has been intensively studied in the community of engineering design (Zhu, 2001). In robust design, the control parameter settings are determined so the system produces the desired mean values for the performance, while at the same time minimizing the variance of the performance (Tay and Taguchi, 1993).

The most commonly applied system design methodology is the top-down procedure from system analysis, proceeding from functional design to detailed design. Within this methodology, robust design is most commonly treated during the detailed design phase. Design for robustness of system topology is normally not considered in this methodology. So the task of robust design is downgraded to parameter tuning and tolerance specification to maintain performance within acceptable limits. Topologically open-ended synthesis by genetic programming provides a way to move robust design forward to the conceptual/functional design stage and thus consider design for robustness from the very beginning, which will augment the current practice of design for robustness in practical design (Hu et al., 2005).

Application of evolutionary computation to robust design has been investigated since the early 1990s and can be classified into three categories (Forouraghi, 2000). The first type applies an evolutionary algorithm to parametric design for robustness. The second type focuses on evolving robust solutions in a noisy environment (Hammel and Back, 1994). A very active area of evolving robust systems is called evolvable hardware (Thompson, 1998). But most of these studies still separate the topology search and parameter tuning.

Two primary approaches to evolution of robust systems have been used by others: Robustness by Multiple Simulation ("RMS") and Robustness by Perturbed Evaluation ("RPE").

A common approach for evolving robust design 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. This GP robust-by-multiple-simulation (RMS) method is used in (Branke, 2001), and in some of the experimental conditions reported here.

Another method is simply to add perturbations to the design variables before evaluation and evaluate a single, perturbed design. The perturbations, however, are not incorporated into the genome, making it different from a “normal” parameter mutation operator or Lamarckian-style evolutionary algorithm. This robust-by-perturbed-evaluation (RPE) method is used in (Tsutsui and Ghosh, 1997) and is suggested to be more efficient by (Jin and Sendhoff, 2003). It is attractive because it uses only a single simulation run for evaluation of each design, relying on the fact that the design will persist and be re-evaluated again in future generations if it is a good one. However, it is not used in the study reported here.

3. Analog Filter Synthesis Using Bond Graphics and Genetic Programming

Bond Graphs

The bond graph is a modeling tool that provides a unified approach to the modeling and analysis of dynamic systems, especially hybrid multi-domain systems including mechanical, electrical, pneumatic, and hydraulic components (Karnopp et al., 2000). The explicit representation of model topology used in bond graphs makes them particularly good candidates for use in open-ended design search using genetic programming – for example, both series and parallel connections appear graphically as trees, unlike their conventional circuit-diagram representation. Complex electrical circuits and mechanical systems, or a synthesis of both, can be modeled as a tree structure using a bond graph, which is easy to evolve with one genetic programming tree. Bond graphs have four embedded strengths for evolutionary design application – namely, the wide scope of systems that can be created because of the multi- and inter-domain nature of bond graphs, the efficiency of evaluation of design alternatives, the natural combinatorial features of bond and node components for generation of design alternatives, and the ease of mapping to the engineering design process. Notation details and methods of system analysis related to the bond graph representation can be found in (Karnopp et al., 2000).

Bond Graph Modeling of Two-Port Active Components

In this section, a common-emitter transistor and operational amplifier are discussed. To represent the operational amplifier and transistor, we will extend the normal bond graph notation, as is frequently done (Karnopp et al., 2000), to include a double arrow to indicate a signal flow, rather than a bond. This allows the definition of controlled sources of effort, which are still one-port components from the bond graph perspective. The GPBG system is able to process the formulation of state equations using signal flows (forcing one of the associated variables to zero) as well as normal bonds.

The state models and parameters of the transistor and op amp components are fairly similar, and they are expected to play very similar roles in a GPBG-modeled circuit, so the initial experiments have been done with either common-emitter transistors or op amps in the designs, but not both. Figure 11-1 is the AC-equivalent-circuit model for the common-emitter transistor, and we can simplify this model to Figure 11-2.

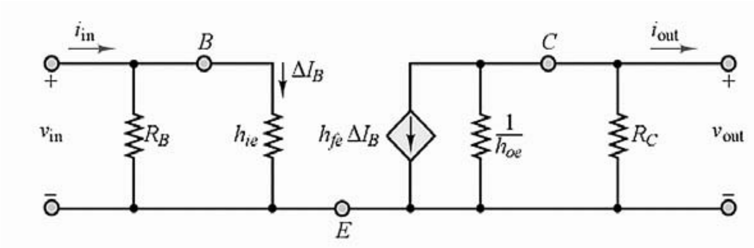

Figure 11-1. AC equivalent-circuit model for the common-emitter transistor

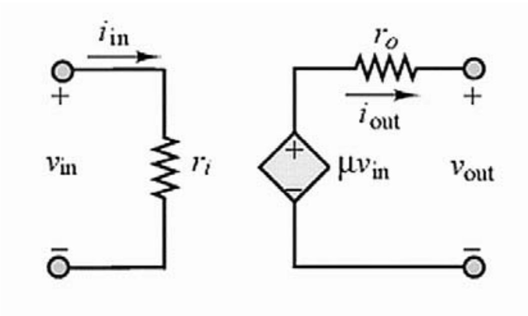

Figure 11-2. Simplified equivalent circuits for the common emitter amplifier

From this simplified circuit of a common emitter amplifier, we can derive, in Figure 11-3, the equivalent bond graph of the circuit of Figure 11-2.

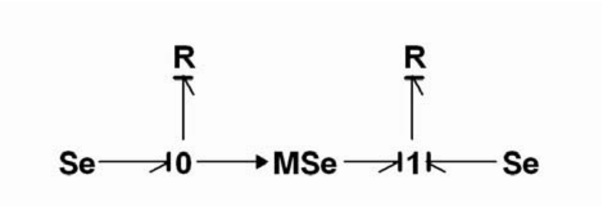

Figure 11-3. Equivalent Bond Graph of Common Emitter Transistor

Again using the signal-flow convention to define a controlled source of effort, we can draw in Figure 11-5 the bond graph equivalent of the circuit of Figure 11-4.

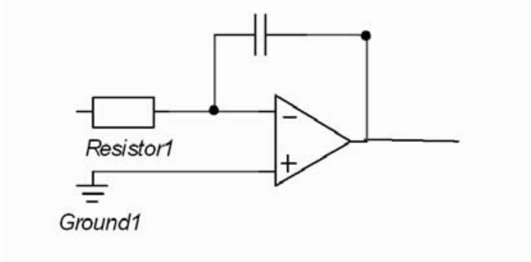

Figure 11-4. Electric Circuits with Operational Amplifier

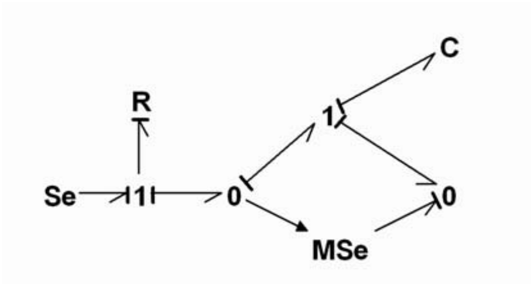

Figure 11-5. Bond Graph Model of Operational Amplifier

In Figure 11-3 and Figure 11-5, the models have a controlled source of effort and we can simplify the models' implementations in GPBG.

Standard components of bond graphs for design of active systems are the inductor (I), resistor (R), capacitor (C), transformer (TF), gyrator (GY), 0-junction (J0), 1-junction (J1), source of effort (SE), source of flow (SF), transistor (TR), and OPAMP. In the electrical context, a source of effort corresponds to a voltage source, and a source of flow, to a current source. For the use of op amps and transistors, we have also included the capability to process signals (represented with full arrowheads) as well as bonds. In this chapter, we concentrate our discussion on active analog filter design, and the resulting bond graphs will be composed of only I, R, C, SE, SF, TR, OPAMP components; however, in this initial study, we will not use both the transistor and operational amplifier in the same bond graph, as the particular (two-port) bond graph transistor model implemented here does not provide capabilities beyond that of the operational amplifier.

Combining Bond Graphs and Genetic Programming

The problem of 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. Building upon Koza's work on automated synthesis of electronic circuits, we created a developmental GP system for open-ended synthesis of mechatronic systems represented as bond graphs. It 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, yielding a final bond graph, and 3) a fitness function to evaluate the performance of candidate solutions. Implicit in the system is the capability to use the bond graph generated to formulate the state equations of the system to allow its simulation or analysis, permitting assessment of its performance (i.e., fitness evaluation).

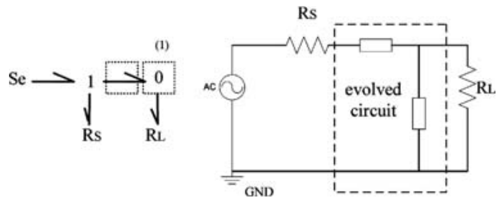

Figure 11-6. Embryo bond graph and its corresponding electric circuit.

In developmental GP, an embryo is used as the root of the GP tree, and is often used to guarantee that each tree contains the minimum structure to allow evaluation of its fitness. For this example, as shown in Figure 6, the embryo assures that each circuit has a voltage source at which the input is applied, a source resistor, and a load resistor across which the output of the filter can be measured. In the GPBG system, the GP tree does not represent the bond graph directly, but is instead a tree-structured program for construction of a bond graph, beginning with the embryo as the root. Figure 11-7 shows a bond graph construction tree, but the details (Fan et al., 2001) are not needed to understand the experiments described here.

Choosing a good function set for bond graph synthesis is not trivial. In our earlier work, we used a very primitive "basic" function set, and later, we developed the following hybrid function set to reduce redundancy while retaining good flexibility in topological exploration:

$$F = \{Insert\_J0E, Insert\_J1E, Add\_C/I/R, EndNode,$$
$$Insert\_Transistor/Insert\_OPAMP, EndBond, ERC\}$$

Figure 11-7 shows a GP tree that specifies how a complete bond graph solution is constructed from the embryo bond graph.

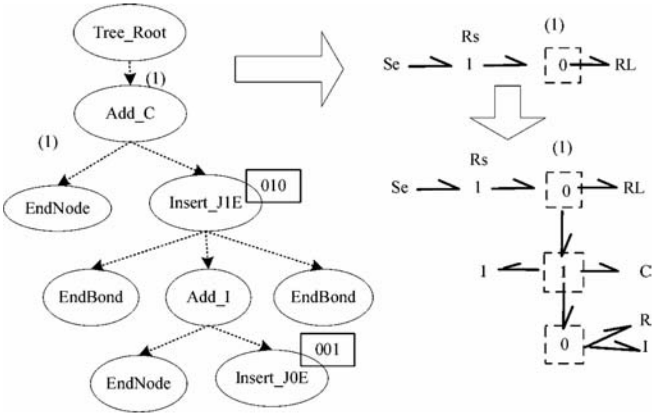

Figure 11-7. A sample GP tree (left), composed of topology operators applied to an embryo (tree_root and bond graph on right), generating a bond graph (lower right) after depth-first execution (numeric nodes omitted).

In this paper, we introduce two new functions: Insert_Transistor and Insert_OPAMP. As was discussed in the previous section, because of the similarity between the common emitter model of the transistor and the operational amplifier, we will use only one generic, active two-port component at a time.

The Example Lowpass Filter Problem

In this study, a lowpass filter with fifth-order Bessel characteristics is to be synthesized. We say a lowpass filter of Bessel characteristics, rather than a Bessel lowpass filter, because a Bessel filter is designed with a strict mathematical equation and a well-defined synthesis procedure, while we simply used the fifth-order Bessel filter magnitude and phase frequency response as reference for design fitness evaluations. Figure 11-8 shows a design of a typical 5th-order Bessel lowpass filter.

In this GPBG-based filter design problem, a bond-graph-represented analog filter composed of capacitors, resistors, inductors, and transistors or operational amplifiers is to be evolved such that the magnitude and phase of its frequency response approximate the Bessel filter frequency response specification. This

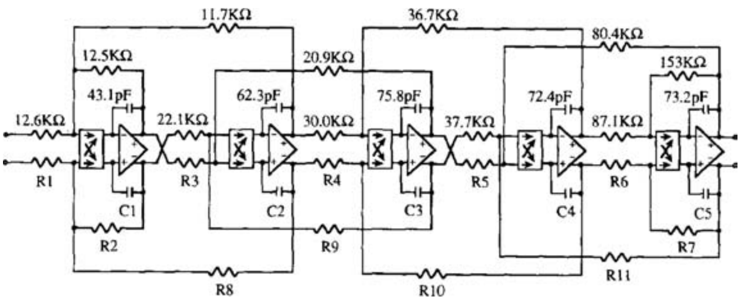

Figure 11-8. 5th-order Bessel Filter.

procedure will not use the sophisticated (and relatively time-intensive) SPICE simulation program, as is typically done in analog circuit analysis. Instead, the frequency response of the circuit modeled by the bond graph can be calculated in a faster and more convenient way: first, the state equation of the bond graph is automatically derived from the model, yielding the A, B, C, and D matrices of linear system theory. The frequency response of the state-space model is then calculated on a Linux PC using C++ simulation code generated by the Matlab 3.0 compiler.

The detailed specifications of the lowpass filter problem addressed here are as follows: the frequency response performance of a candidate filter is defined as the weighted sum of deviations from ideal magnitude and phase frequency responses evaluated at 101 points:

$$F_{magnitude}(t) = \sum_{i=0}^{100}[W(d(f_i), f_i) * d(f_i)] \tag{11.1}$$

$$F_{phase}(t) = \sum_{i=0}^{100}[W(d(f_i), f_i) * d(f_i)] \tag{11.2}$$

The definition of the frequency response magnitude is the same as in our earlier study [Hu, 2004]. However, in this study, we also include frequency response phase in the calculation of fitness, where f_i is the sampled frequency, $d(x)$ is the absolute deviation of candidate frequency response from target response at frequency x, and $W(x, y)$ is the weight function specifying the penalty level for a given frequency response at a specified frequency range. The sampling points range from 1Hz to 100 KHz, evenly distributed on a logarithmic scale. If the deviation from ideal phase is less than 30 degrees, the weight is 1. If the deviation is more than 30 degrees, the weight is 10, aimed at reflecting the relatively small importance of small deviations in the phase response from ideal. The pass band is [1, 1k] Hz, the stop band is [2K, 10K] Hz. The phase is weighted as 0.1 and magnitude is weighted 0.9 in the final fitness calculation. An alternative could be to treat magnitude and phase as separate objectives in a multi-objective search, but our design here is simple and produces an acceptable result. If we want to have stringent control of the phase, we need to consider other alternatives, but that was not seen as critical to this study.

Before introduction of any robustness considerations, the fitness function is defined as follows. First we calculate the raw fitness defined as the average absolute deviation between the frequency response magnitude and phase of the candidate solution and the target frequency response over all 101 sampling frequencies.

$$f_{raw} = \frac{1}{101}(0.9 * f_{magnitude} + 0.1 * f_{phase}) \tag{11.3}$$

$$f_{norm} = \frac{NORM}{NORM + f_{raw}} \tag{11.4}$$

Differences from the Usual GP System

The GPBG system includes the following “non-standard” features:

- A flag bit mutation operator is introduced to evolve the configuration of C/I/R elements attached to a junction. That is, junctions introduced into a bond graph by the Insert_J0E or Insert_J1E operators may each have zero or one C, I, and R elements, as specified by three binary flags (rectangles in Figure 11-2 are an example). A special mutation operator can manipulate those flag bits at the junctions.

- A subtree-swapping operator is used to exchange non-overlapping subtrees of the same individual (GP tree).

- A Gaussian ERC mutation operator, as is commonly used in evolution strategies, is developed to evolve the parameter values of all C/I/R components; the value generated for the ERC is arbitrary within the range specified for each component.

Elitism of one individual is used throughout the evolution process – that is, the best individual in the population is always preserved to the next generation.

Except for the above, the GPBG system as used in this paper is a standard strongly-typed multi-population generational GP. The running parameters are specified in Section 5.

4. Evolving Robust Active Analog Filters Using Bond Graphs and Evolutionary Algorithms

This section examines the design of analog filters for robustness to parameter variations, There are many ways this might be attempted using GP as an open-ended topological search tool.

GPRMS

For the GPRMS multi-simulation method, the raw fitness for a design solution, including a robustness criterion, is defined as the sum of a number NS (here, 10) of (here, 101-point-) deviation sums from the target frequency response curve, resulting from NS filter simulations of the same design:

$$f_{robustraw} = \sum_{k=1}^{NS} f_{raw}^{k} \tag{11.5}$$

where NS is the number of Monte Carlo sampling evaluations (filter simulations) for each individual, and f is the raw fitness of the kth sampled evaluation with a different Monte Carlo perturbation of the parameters, as defined in Equation 11.3. With this raw robustness from Equation 11.5, we then calculate the final fitness similarly to Equation 11.4

5. Experiments and Results

In this section, a series of experiments is conducted to verify the effectiveness of introducing active components into robust design of an analog filter by genetic programming. In these experiments, the perturbation of the component values during evolution is implemented by adding to each component's parameter(s) Gaussian noise $N(\mu, \sigma)$ with mean $\mu = 0$ and standard deviation σ set at 20% of the parameter value. This perturbation model has been widely used in previous research, and while it may not be an accurate model of component variation (introducing more than is typically present in the components), any excess noise may also be useful in discovery of robust solutions. One difficulty with this definition is that if the original parameter value is zero, then no perturbation will be generated. Although this is rare in evolutionary experiments, it is alleviated here by checking for any component value of zero, in which case the standard deviation for the perturbation is set to 1.0.

In the robust design of GPRMS (using the robustness by multiple simulations approach), multiple simulations (in this case, NS = 10) are used to evaluate the fitness of each single design. For this filter design problem, the computation budget is 1,000,000 simulations, so up to 100,000 different designs can be evaluated in each run. Runs in which best performance fails to advance for X generations are terminated automatically with fewer than 100,000 individuals evaluated (i.e., fewer than 1,000,000 simulations).

While only one parameter perturbation model was used during the evolutionary synthesis experiments – Gaussian noise $N(\mu, \sigma)$ with mean μ of 0 and standard deviation σ set at 20% of the parameter value, the later (post-run) robustness evaluations of the evolved filters include multiple perturbation magnitudes with extensive simulation.

To assess the statistical significance of the performance differences between these methods, 15 runs were done for each synthesis method. The size of these experiments was determined by the computing resources available. However, since the results were found to be quite stable across multiple runs, this level of replication appears to be sufficient for the purpose of this preliminary study.

All experiments described below used the same embryo bond graph shown in Figure 11-6. The component values of source resistor R_s and load resister R_{load} are both 1Ω for the lowpass filter with Bessel characteristics.

The following sections first describe separately the experimental configuration of each method, the best evolved bond graph model of the filter, and the magnitude and phase responses of the best solution from each method. These results provide some general ideas regarding how robustness is evolved with respect to the parameter perturbations. Then a statistical comparison of the performance is presented. The following common running parameters (Table 11-1 were used throughout all GP experiments in this chapter.

Table 11-1. Shared parameters of experimental runs.

Total population size: 2000	Crossover probability: 0.4
MaxDepth: 10	Standard mutation probability: 0.05
InitDepth: 3-5	Parametric mutation probability: 0.3
Tournament size: 2	Flag mutation probability: 0.3

At the end of the run, the robustness of each final evolved solution was evaluated against a series of perturbation magnitudes: Gaussian noise $N(\mu, \sigma)$ with mean μ at 0 and standard deviation σ at 10% to 50% of parameter values, in steps of 10%, each tested with 5000 samplings with different configurations of the component parameter perturbations.

Below, we display the evolved filter with the highest performance from each of the three run types (passive without perturbations during evolution, passive with perturbations during evolution, and active with perturbations during evolution), to test its noise tolerance in the face of degradation or variation of the component parameters.

Evolving Robust Passive Analog Filters Using Genetic Programming: Open-Ended Topology Innovation for Robust Design

In the experiments, analog lowpass filters with 5th-order Bessel characteristics were evolved using GPRMS and incorporating a robustness criterion (Equation 11.5) in the fitness function Equation 11.3, in two ways: 15 were evolved using only passive components and 15 were evolved also allowing active components – either common-emitter-modeled transistors or op amps.

From Figure 11-9 first, one can see that the frequency response magnitude in the plot of the filters evolved, when evaluated according to the criterion of Equation 11.1, resembles the target, but appears to represent a higher-order filter (with a sharper falloff). It is expected that further evolution might produce characteristics more similar to those of the 5th-order Bessel filter. The observation is that introducing a robustness requirement does not necessarily decrease the performance with nominal parameters significantly. However, the phase response deviates noticeably, particularly at high frequency. A possible explanation is that the weight of the phase term in the fitness function was only 10%, compared to 90% for the magnitude term. This weighting was used to reflect the differential importance of amplitude and phase for most applications. From the perspective of system performance, the phase delay is not usually a critical objective in the real application, often being set instead as a constraint with no effect on fitness so long as it does not fall outside a specified range. However, its performance is contrasted with that of GP and in the statistical comparisons below.

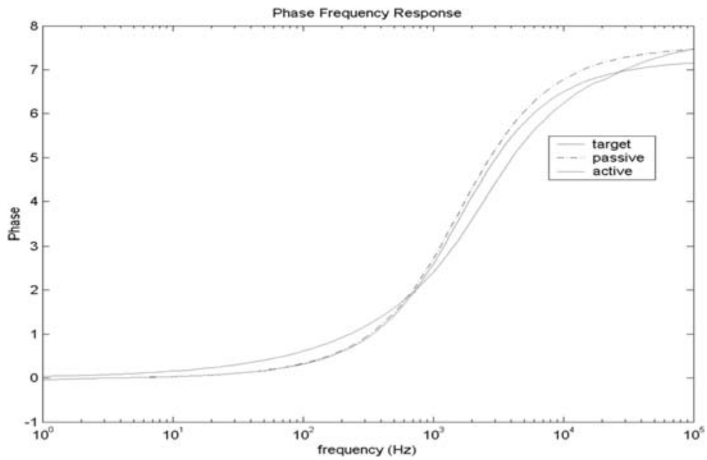

Figure 11-9. Magnitude and Phase Frequency Responses.

Evolving Robust Active Analog Filters

In the second set of runs, robust analog filters with active components are evolved that have higher tolerance to the variations of component values. Figure 10 shows the best filter evolved with only passive components and Figure 11 shows the best evolved with active components. This filter uses far fewer components than the filter evolved only using passive components, while its functional performance remains similar. The robustness of this filter is next compared to that of the best filter evolved with only passive components.

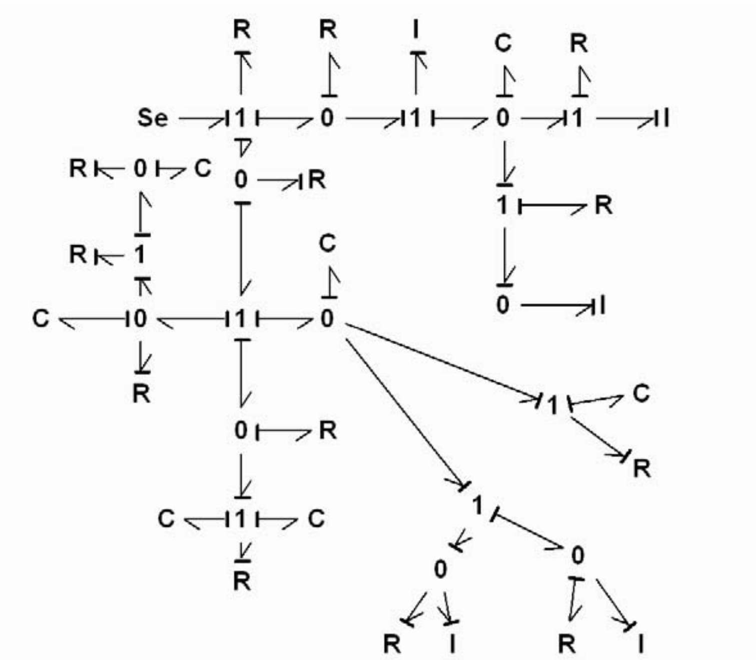

Figure 11-10. The best gond graph evolved with GPRMS and only passive components.

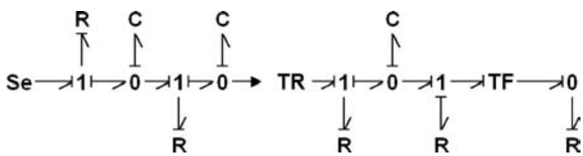

Figure 11-11. The best bond graph evolved with GPRMS with active components available

Statistical Comparisons of the Three Methods

Hypothesis 1: (re passive filters) Systems with similar "base" (unperturbed) performance can be evolved if robustness is considered during the evolutionary process, without the need for a much larger number of function evaluations than is needed for conventional (non-robust) synthesis. A t-test on the differences in performance of the filters evolved by GPRMS and by plain GP, evaluated with their nominal (unperturbed) parameter values, did not reveal any significant difference (e.g., $P > 0.20$). They were evolved using an identical number (1 million) of filter simulations.

Hypothesis 2: (re passive filters) Introduction of noise during the process of evolution improves the robustness over the "plain GP" results: a t-test on the results of the lowpass filter problem was used to compare the robustness of the evolved solutions by GPRMS and standard GP in terms of fitness at the 0.2 (20%) perturbation level. A significance level of $P <= 0.001$ was achieved; strongly indicating that GPRMS improved the robustness over the filters evolved by plain GP, using the same number of filter evaluations.

Hypothesis 3: (for active vs. passive filters) Introduction of active components during the process of evolution allows the circuit size to decrease without loss of performance or robustness. A t-test was done on the results from the lowpass filter with passive components and active components to compare the robustness of the evolved solutions in terms of fitness at the 0.2 perturbation level. It did not reveal any significant difference (e.g., $P > 0.20$), which indicates that the decrease in size (number of components) observed for the active filters did not decrease the robustness of the filters evolved.

6. Conclusions and Future Work

This chapter exploits the open-ended topological search capability of genetic programming to conduct preliminary studies of robust design of dynamic systems with active components. The common emitter transistor model and operational amplifier are represented using bond graphs with a signal-flow extension, and are implemented as new components in the GPBG system for evolutionary design. This extended system is used to test hypotheses involving the use of topological innovation in the conceptual design stage to improve the robustness of the systems evolved. Specifically, GPBG in the RMS (robustness by multiple simulations) framework is used to design analog filters of high robustness. Evolving robustness is a rich research theme and there are several interesting topics to be further investigated. This experiment is the only the first step enabled by addition of active components to the GPBG system, broadening the scope of design problems for which it can be used.

References

Branke, J. (2001). *Evolutionary Optimization in Dynamic Environments.* Kluwer.

Carlson, J. M. and Doyle, J. (2002). Complexity and robustness. In *Proceedings of National Academy of Science (PNAS)*, volume 1, pages 2538–2545.

Du, X. and Chen, W. (2000). Towards a better understanding of modeling feasibility robustness in engineering design. *ASME J. Mech*, 122(4):385–394.

Fan, Zhun, Hu, Jianjun, Seo, Kisung, Goodman, Erik D., Rosenberg, Ronald C., and Zhang, Baihai (2001). Bond graph representation and GP for automated analog filter design. In Goodman, Erik D., editor, *2001 Genetic and Evolutionary Computation Conference Late Breaking Papers*, pages 81–86, San Francisco, California, USA.

Hammel, U. and Back, T. (1994). Evolution strategies on noisy functions: how to improve convergence properties. In *Proceedings of Parallel Problem Solving from Nature*, volume 3, pages 159–168.

Hu, J., Goodman, E.D., Seo, K., fan, Z., and Rosenberg, R. (2005). The hierarchical fair competition (hfc) framework for sustainable evolutionary algorithms. *Evolutionary Computation*, 13(1).

Jin, Y. and Sendhoff, B. (2003). Trade-off between optimality and robustness: An evolutionary multi-objective approach. In Fonseca, C., editor, *In Proceeding of the Second Int. Conf. on Evolutionary Multi-Criterion Optimization*, pages 237–251. Spinger.

Karnopp, D.C., Margolis, D.L., and Rosenberg, R.C. (2000). *System Dynamics: Modeling and Simulation of Mechatronic Systems*. John Wiley and Sons, Inc., New York.

Koza, John, Keane, Martin, Streeter, Matthew, Mydlowec, William, Yu, Jessen, and Lanza, Guido (2003). Routine human-competitive machine intelligence. In *Genetic Programming IV*. Kluwer Academic Publishers.

Tay, E. and Taguchi, W. (1993). *Taguchi on Robust Techology Development: Bringing Quality Engineering Upstream*. American Society of Mechanical Engineering Press, New York.

Thompson, A. (1998). On the automatic design of robust electronics through artificial evolution. *International Conference on Evolvable Systems*, pages 13–24.

Tsutsui, S. and Ghosh, A. (1997). Genetic algorithms with a robust solution searching scheme. *IEEE Trans. Evolutionary Computation*, 1(3).

Zhu, J. (2001). Performance distribution analysis and robust design. *Journal of Mechanical Design*, 123(1):11–17.