Fuzzy controller implementations with fewer than ten transistors?

Adrian Stoica Ricardo S. Zebulum Didier Keymeulen
M. I. Ferguson Vu Duong

Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109
818-354-2190
adrian.stoica@jpl.nasa.gov

Abstract - Evolutionary algorithms (EA) offer good promise for automated design of analog circuits as well as for adaptation and automatic reconfiguration of programmable devices. In particular, EAs facilitate the design of analog circuits for very specific requirements, such as those related to the implementation of fuzzy operators, or even of complete fuzzy systems. The paper starts with a brief overview of the evolutionary process applied to circuit design and of a family of analog programmable devices that support on-chip evolution. As a case study, we describe the evolutionary design of a fuzzy controller, using re-configurable analog chips models and unstructured representation. We were able to achieve a circuit that approximates the control surface of a 2-input fuzzy controller, mapping thus a full fuzzy system in only seven transistors. The paper presents evidence that EA can provide very compact solutions for implementation of fuzzy systems, and that programmable analog devices are an efficient and rapid solution for rapid deployment of fuzzy systems.

I. INTRODUCTION

Evolutionary algorithms (EA) offer good promise for automated design of analog circuits as well as for adaptation and automatic reconfiguration of programmable devices. A variety of computational analog circuits and filters [1,2,3], as well as digital circuits [4] have been synthesized by EAs. On the other hand, used with reconfigurable devices in the loop, EAs were able to perform the configuration search directly in hardware, e.g. in experiments using Field Programmable Gate Arrays (FPGAs) [5,6] or Field Programmable Transistor Arrays (FPTAs) [7], leading to device configurations (designs) that satisfied the imposed requirements. More details on current work in evolvable hardware can be found in [8] and [9].

This papers focuses on the application of EA to the design of a specific category of circuits, i.e. circuits used in the implementation of fuzzy systems. It presents both circuit solutions that could be fabricated as Application Specific Integrated Circuits (ASICs) and circuit topologies that can be mapped on programmable devices, such as the FPTAs. The claim is that EAs offer the ability of automatic design of very compact circuits that can approximate complete fuzzy systems.

The paper is organized as follows: Section 2 provides a brief overview of the main concepts of evolutionary design of electronic circuits. Section 3 reviews the FPTA architecture used as experimental platform for evolutionary experiments and support for the rapid implementation of evolved fuzzy circuits. Section 4 describes the particular problem being tackled. Section 5 illustrates how the FPTA can be used to evolve reconfigurable circuits implementing complete fuzzy systems. Section 6 presents a very compact evolved circuit that approximates the target fuzzy system, using an unstructured circuit representation. Finally, section 7 concludes the work.

II. EVOLUTIONARY SYNTHESIS OF ANALOG CIRCUITS

The evolution of electronic circuits is based on a population of competing designs, the best ones (i.e. the ones that come closer to meeting the design specifications) being selected for further investigation. Each candidate circuit design is associated with a "genetic code" or chromosome.

The simplest representation of a chromosome is a binary string, a succession of 0s and 1s that encode a circuit. The first step of evolutionary synthesis is to generate a random population of chromosomes. The chromosomes are then converted into a model that gets simulated (e.g. by a circuit simulator such as SPICE) and produces responses that are

0-7803-7282-4/02/$10.00 ©2002 IEEE

compared against specifications. Or, the chromosomes are transformed into a configuration bitstring downloaded into a programmable device.

The configuration bitstring determines the functionality of the cells of the programmable device and the interconnection pattern between cells. Circuit responses are compared against specifications of a target response and individuals are ranked based on how close they come to satisfying it.

We can devise two methods to represent circuits using SPICE netlists: using models of programmable devices or using an unstructured representation. In the former, a binary representation is employed to provide the state of the switches of the configurable device, as detailed in the next section; in the latter, an integer representation is used to map a circuit, as detailed next.

The unstructured representation establishes a straightforward mapping between the electronic circuit topology and the integer strings processed by the GA. Each functional block of the string, also called gene, states the nature, value, and connecting points of a correspondent electronic component, which may include resistors, capacitors, bipolar transistors and MOS (Metal-Oxide-Semiconductor) transistors.

Figure 1 depicts an example of this kind of chromosome-circuit mapping for a common emitter amplifier.

The chromosomes are made up of genes, each of which encodes a particular component. In the example of Figure 1, the chromosome will consist of three genes. The gene determines the nature, value and connecting points of the related component. The total number of connecting points is a parameter to be set in this representation. This parameter is critical to the efficiency of the representation: if too few connecting points are considered, the number of possible topologies sampled by the evolutionary algorithm will be limited; conversely, if too many connecting points are considered, a higher number of non-simulatable topologies (with floating components) will arise. Additionally, each connecting point may be classified as internal or external. While the former does not serve for any special purpose, the latter is connected to one of the following signals: power supply, ground, input signal or probed output (Figure 1).

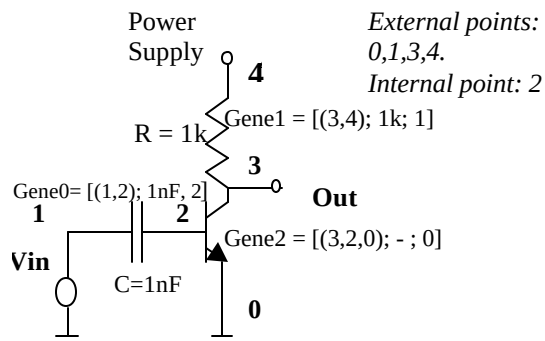

Figure 1– Analog Circuit's Representation. Gene = [Connecting points, Component value,Component nature]. The Component nature is given by: 0 = transistor; 1 = resistor; 2 = capacitor

The advantage of this representation is its flexibility to map circuits with arbitrary types of interconnections, and, opposing to other representations, there is no bias towards the evolution of well known topologies. In this particular article the components used by the EA are NMOS and PMOS transistors.

III. FIELD PROGRAMMABLE TRANSISTOR ARRAYS

Evaluation of a circuit directly on a programmable device may offer a substantial advantage in circuit evaluation time; in certain cases the time for hardware evaluation can be seconds instead of days, as often the case when evaluation is in software. Additionally, one can also evolve a circuit using SPICE netlists of the programmable array, and download the final result of evolution onto the chip.

The FPTA is a concept design for hardware reconfigurable at transistor level introduced in [2]. As both analog and digital CMOS circuits ultimately rely on functions implemented with transistors, the FPTA appears as a versatile platform for the synthesis of both analog, digital and mixed-signal circuits. The architecture is cellular, and has similarities with other cellular architectures as encountered in FPGAs (e.g. Xilinx X6200 family) or cellular neural networks. One key distinguishing characteristic relates to the definition of the elementary cell, an example of cell being shown in Figure 2. The architecture is largely a "sea of transistors" with interconnections implemented by other transistors acting as signal passing devices (gray-level switches), and with islands of RC resources in between.

0-7803-7282-4/02/$10.00 ©2002 IEEE

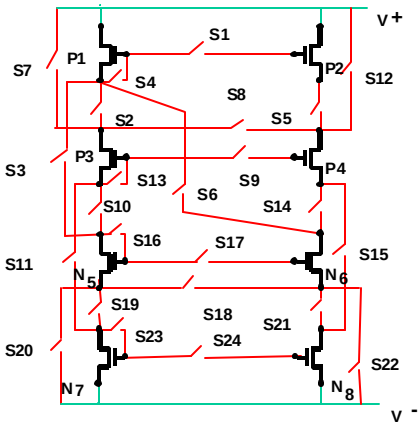

Figure 2 – FPTA cell topology.

The status of the switches (ON or OFF) determines a circuit topology and consequently a specific response. Thus, the topology can be considered as a function of switch states, and can be represented by a binary sequence, such as "1011…", where by convention one can assign 1 to a switch turned ON and 0 to a switch turned OFF. Programming the switches ON and OFF defines a circuit for which the effects of non-zero, finite impedance of the switches can be neglected in the first approximation (for low frequency circuits).

IV. EVOLUTION OF ANALOG CIRCUIT APPROXIMATIONS OF COMPLETE FUZZY SYSTEMS

The evolution of analog controllers is a promising path for showing the potential of evolutionary electronics applied to potential industrial applications. Particularly, this case study refers to the evolution of circuits implementing a complete fuzzy system. The reader also can refer to [10], where EAs were applied to determine circuits that implement operators such as conjunctions and disjunctions modeled by triangular norms, which is a central element for fuzzy systems.

Most of the traditional fuzzy systems in use however, are quite simple in nature and the computation can be expressed in terms of a simple surface. An example is the control surface of a two-input fuzzy controller. A fuzzy circuit could be synthesized to approximate this surface.

The example chosen is that of a fuzzy controller provided as a demo for the popular MATLAB software [11]. The "ball juggler" is one of the demos of the MATLAB Fuzzy Logic Toolbox. The fuzzy controller for the ball juggler has two inputs and one control output. A screen capture illustrating the membership functions is shown in Figure 3. The controller is a simple Sugeno-type with 9 rules. A screen capture of the control surface is shown in Figure 4.

Next we describe two approaches to evolve this control surface, using the FPTA model and using an unstructured representation.

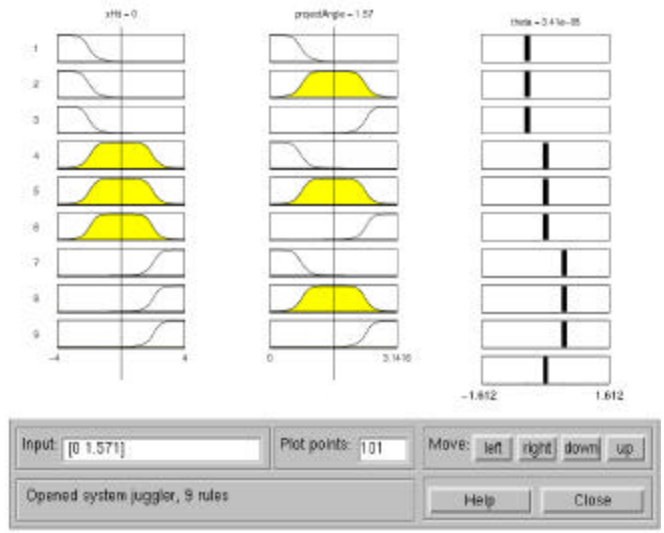

Figure 3 – Membership functions for the ball-juggler fuzzy controller.

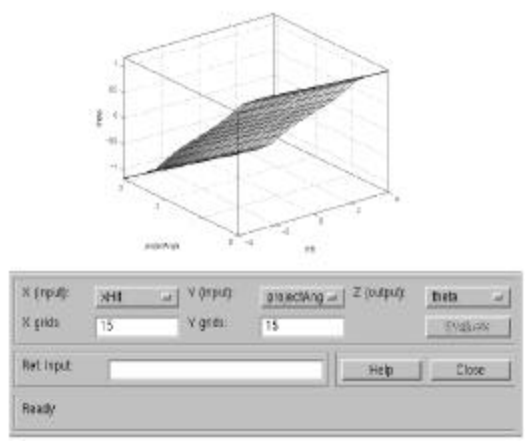

Figure 4 – Surface of the ball juggler fuzzy controller.

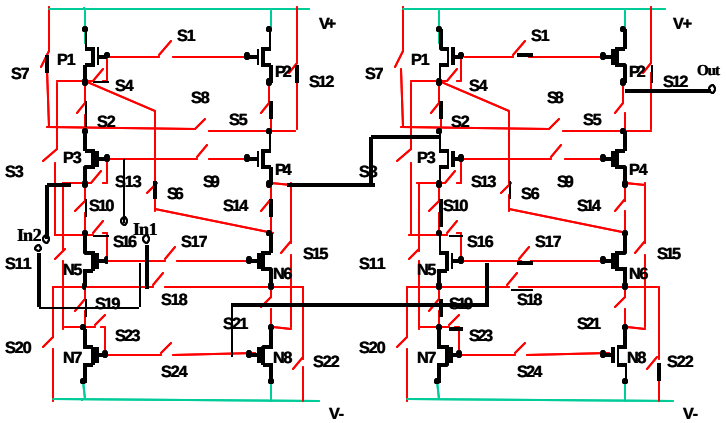

Figure 5 – Evolved Circuit using SPICE model including 2 FPTA cells for the ball-juggler fuzzy controller.

V. EVOLUTION USING THE FPTA MODEL

The fuzzy control surface described in the last section has been evolved using a SPICE model for the FPTA cell shown in Figure 2. Two cells were cascaded and the Evolutionary Algorithm evolved the states of the ON/OFF switches.

The EHWPack software [12] was used in this experiment. EHWPack is a distributed parallel software-hardware environment for evolutionary circuit design. It runs on the 128 nodes Origin 2000 Scalable Shared Memory Multiprocessor system from Silicon Graphics, and is remotely controlled from a local workstation. It has been developed to facilitate the experiments in simulated and hardware evolution using SPICE circuit simulation and the FPTA. The tool is used for the evolutionary synthesis and optimization of electronic circuits.

In this particular experiment, the Evolutionary Algorithm processed 128 individuals along 200 generations. The experiment lasted around 10 minutes when using 16 processors of the machine. The circuit depicted in Figure 5 was evolved. Figure 6 compares the attained with the target response.

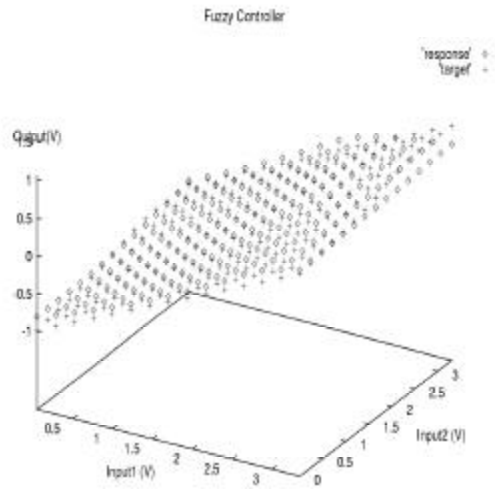

Figure 6 - Simulated response of a circuit implementing the ball-juggler fuzzy controller (?). Target characteristic shown with (+). x,y axis are for inputs, z (vertical) is the output. Axes are in Volts.

0-7803-7282-4/02/$10.00 ©2002 IEEE

This circuit operates in voltage mode. In order to evaluate the performance, we use the relative error given by the ratio between the absolute error and the difference between the maximum and minimum values of the fuzzy control surface. The average error to the target was of 3.49%, and the highest error was of 13.3%. Figure 7 shows the fitness of the best individual along the generations. The fitness was made up of two terms, one giving the Mean Squared Error to the target and another giving the highest error. They were aggregated after being normalized.

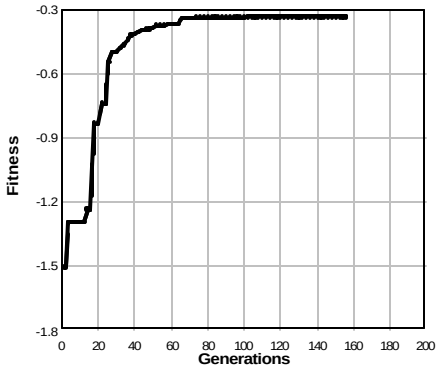

Figure 7 – Fitness along the generations for the ball-juggler fuzzy controller .

VI. EVOLUTION USING UNSTRUCTURED REPRESENTATION

A compact circuit, using only seven transistors and operating in current mode, was evolved. This circuit, which approximates more accurately the control surface than the previous one, is depicted in Figure 8. The response, presented together with the target surface for comparison is shown in Figure 9. The average error achieved was of 1.93%, and the maximum error to the target surface was 6.7%.

The circuit is rather robust, and was tested at variations in transistor sizes, supply voltage and temperature, with the following results: decreasing the transistor sizes by a factor of 10 did not change the circuit response and the deviation from the target; average error of 1.98% and maximum error of 6.96% when decreasing the power supply voltage to 4.75V; average error of 1.94% and maximum error of 6.65% when increasing the power supply voltage to 5.25V; average error of 1.89% and maximum error of 6.3% when decreasing the temperature to 0°C; average error of 1.98% and maximum error of 7.2% when increasing the temperature to 55°C.

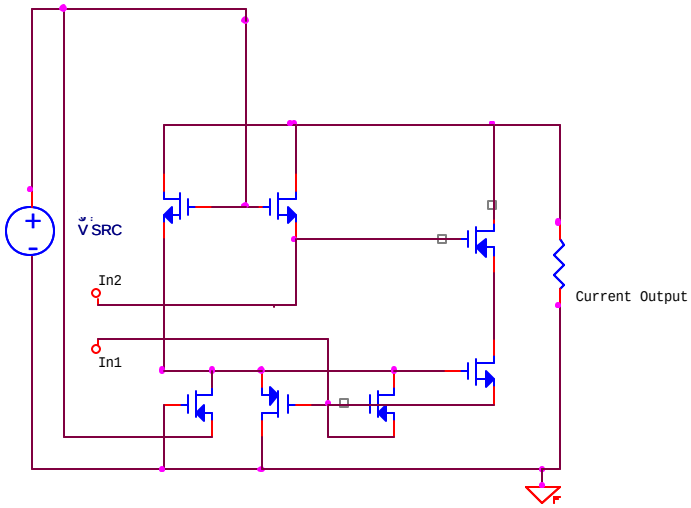

Figure 8 – Evolved circuit realizing ball juggler fuzzy controller. Transistor substrate connections at 5V for PMOS and 0V for NMOS. VSRC at 5V; Inputs In1 and In2; Current output probed at the resistor (10k).

Finally, a different model, (specific for a HP 0.5 MOS fabrication process) led to qualitatively the same result, with slight increase in the error. That error became small again when evolution targeted a circuit in that specific process.

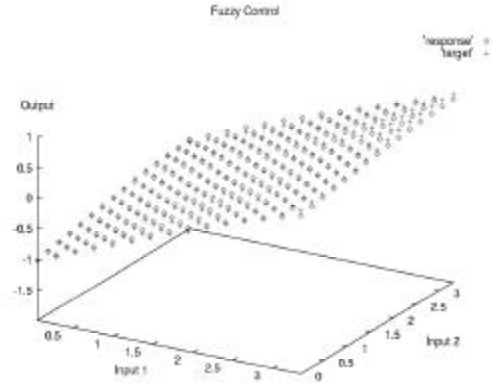

Figure 9 – Comparison between response and target for the evolved fuzzy controller.

VII. CONCLUSIONS

One can look at circuits with transistors as functional approximators –how many transistors are needed is only a question of how accurate a function needs to be. Two main results are suggested by the experiments in this paper: that very compact solutions for complete fuzzy systems can be custom designed by evolutionary algorithms,

0-7803-7282-4/02/$10.00 ©2002 IEEE

and that specific programmable analog devices could be used as a general purpose platform to rapidly prototype and deploy any fuzzy system. Thus, the programmable analog device solution comes between the high-performance (in speed and power) but very expensive and inflexible full ASIC solution and the less-performance but cheaper and flexible microprocessor solution.

The netlists of the circuits shown in this paper will be soon be available at http://cism.jpl.nasa.gov/ehw/public/cec02.

Acknowledgements

The work described in this paper was performed at the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the National Aeronautics and Space Administration.

References

[1] Koza J. R., Bennett III F. H., Andre, D. , Keane, M. A., "Four Problems for which a Computer Problem Evolved By Genetic Programming is Competitive with Human Performance", Proc. of 1996 IEEE International Conference on Evolutionary Computation, IEEE Press., pp. 1-10.

[2] Stoica., A., "Toward evolvable hardware chips: experiments with a programmable transistor array." Proceedings of 7th International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems, Granada, Spain, April 7-9, IEEE Comp Sci. Press, 1999. pp 156-162.

[3] Thompson, A. , Layzell , P., Zebulum, R.S., "Explorations in Design Space: Unconventional Electronics Design Through Artificial Evolution", published at: *IEEE Transactions on Evolutionary Computation*, Special Issue in "Evolvable Hardware", Moshe Sipper (Ed), pp.167-196, V.3, N.3, September, 1999.

[4] Zebulum, R., Pacheco, M.A., Vellasco, M., "Evolutionary Systems Applied to the Synthesis of a CPU Controller", Second Asia-Pacific Conference on Simulated Evolution and Learning, Canberra, SEAL98, Australia, November, Lecture Notes in Artificial Intelligence, Springer Verlag, 1998.

[5] Thompson, A., Harvey, I., Husbands, P., "Unconstrained Evolution and Hard Consequences", em *Towards Evolvable Hardware: An International Workshop*, Lausanne, Swiss, in Towards Evolvable Hardware: The evolutionary engineering approach, pp. 136-165, E. Sanchez and M. Tomassini (eds), Springer-Verlag LNCS 1062, 1996.

[6] Higuchi , T. et al. "Real-World Applications of Analog and Digital Evolvable Hardware." In IEEE Transactions on Evolutionary Computation, Vol.3, No. 3, September 1999. IEEE Press.pp.220-235.

[7] Stoica , A., Keymeulen , D., Tawel , R., Salazar-Lazaro, C., Li, W., "Evolutionary Experiments with a Fine-Grained Reconfigurable Architecture for Analog and Digital CMOS Circuits." In Proceedings of the First NASA/DoD Worshop on Evolvable Hardware, pages 76-84. IEEE Computer Society Press, 1999.

[8] Stoica, A., Zebulum, R., and Keymeulen,,D. "Progress and Challenges in Building Evolvable Devices", Third NASA/DoD Workshop on Evolvable Hardware, Long Beach, July, 12-14, 2001, pp.33-35, IEEE Computer Society.

[9] Stoica, A., Keymeulen, D., and Zebulum, R., "Evolvable Hardware Solutions for Extreme Temperature Electronics", Third NASA/DoD Workshop on Evolvable Hardware, Long Beach, July, 12-14, 2001, pp.93-97, IEEE Computer Society.

[10] Stoica, A., Zebulum, R., Keymeulen, D., Jin, Y., Daud, T., "Evolutionary Design of Smart Analog Circuits". In Proceedings of ANNIE2000 (Smart Engineering System Design), St. Louis, MO, November 5-8, 2000, ASME press, pp. 245-250.

[11] http://www.mathworks.com.

[12] Keymeulen, , D., Klimeck, G., Zebulum, R., Stoica, A., and Lazaro, C. "EHWPack: A Parallel Software/Hardware Environment for Evolvable Hardware." In Whitley Darrell (eds.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000), July 8-12, 2000, Las Vegas, Nevada USA. San Francisco, CA: Morgan Kaufmann.

0-7803-7282-4/02/$10.00 ©2002 IEEE