Towards Evolvable Analog Artificial Neural Networks Controllers

J.F.M. Amaral, J.L.M. Amaral
Electronics Engineering Department
UERJ - Rio de Janeiro State University
R. São Francisco Xavier, 524, Maracanã,
Rio de Janeiro, RJ, Brazil, 20550-013
[franco, jamaral]@uerj.br

C. Santini, R. Tanscheit, M. Vellasco, M. Pacheco
ICA : Computational Intelligence Laboratory
Electrical Engineering Department
PUC-Rio – Catholic University of Rio de Janeiro
R. Marques de S. Vicente, 225, Gavea,
Rio de Janeiro, RJ, Brazil, 22453-900
[santini, ricardo, marley, marco]@ele.puc-rio.br

Abstract

This work deals with the design of analog circuits for Artificial Neural Networks (ANNs) controllers using an Evolvable Hardware (EHW) platform. ANNs are massively parallel systems that rely on simple processors and dense arrangements of interconnections. These networks have demonstrated their ability to deliver simple and powerful solutions in several areas, including control systems. The EHW analog platform is a reconfigurable platform, called Programmable Analog Multiplexer Array-Next Generation (PAMA-NG), which can be programmed by Genetic Algorithms to synthesize circuits. This article focuses on the development of artificial neuron circuits for analog ANNs on the PAMA-NG.

1. Introduction

The interest in the use of Evolvable Hardware techniques [1] in the design of intelligent analog electronic circuits [2][3] has shown a considerable increase lately. In the last decades, Artificial Neural Networks (ANNs) [4] have been successfully applied to many different problems in the areas of function approximation, pattern recognition and system identification and control: non-linear mapping, learning from examples, parallelism, generalization and robustness (fault tolerance). Especially in system identification and control, ANN applications have achieved valuable results.

The most common implementations of ANNs controllers rely on microprocessors. However, ANNs are massively parallel [4] and, therefore, an analog implementation could be preferable. Since the design of analog circuits is not an easy task, Evolvable Hardware (EHW) techniques have been of great help in programming reconfigurable platforms in order to carry out circuit design [5][6][7]. Reconfigurable platforms consist of integrated circuits where internal connections can be programmed by the user. Field Programmable Analog Arrays (FPAs) constitute the state of the art in the technology of reconfigurable analog platforms [8]. The reconfiguration of these devices enables the self-adapting and self-repairing features, which are important in applications where circuits need to operate for long periods in harsh environments. This is especially appealing to ANNs controllers hardware that could be used in this type of applications.

The EHW analog platform called Programmable Analog Multiplexer Array-Next Generation (PAMA-NG) [9] is a reconfigurable platform consisting of integrated circuits of which the internal connections can be programmed by Evolutionary Computation techniques – such as Genetic Algorithms (GAs) – to synthesize circuits. The PAMA-NG is classified as a Field Programmable Analog Array (FPAA), with features of self-adaptation and self-repair by means of automatic reconfiguration. In the PAMA-NG this is performed by GAs.
Traditionally, analog hardware implementations of ANNs are based on designer’s experience and intuition following electronic circuit design common rules [10]. On the other hand, synthesis of unconventional electronic circuits is particular appealing to EHW. Its use in the synthesis of analog electronic circuits that implement ANNs has been explored recently [11][12].

This work deals with the implementation of ANN controllers based on building blocks (artificial neurons) developed on PAMA-NG and is divided into four additional sections. The next one discusses some architectural aspects of ANNs and their analog hardware implementations. Section 3 describes the PAMA-NG (Programmable Analog Multiplexer Array – Next Generation) [9] and the procedure carried out to evolve artificial neuron circuits. Section 4 presents preliminary experiments and Section 5 concludes the work.

2. Artificial Neural Networks Controllers

Artificial Neural Networks can model highly complex non-linear functions. When modelling a Multi-Input Single-Output (MISO) system, an ANN maps the n-dimensional input space to the single dimension output space. In control applications, inputs are usually the error (actual output minus setpoint) and change in error (difference between the current and previous errors), and the output is the input to the plant to be controlled. In this case, the ANN implements the complex mapping function needed to control the plant.

An ANN consists of a network (Figure 1) of highly interconnected, simple processing elements called neurons (Figure 2).

![Figure 1. Artificial Neural Network](image1)

The mathematical model of a single neuron is given by the following expression:

\[ y_k = \varphi \left( \sum_{j=1}^{m} w_{kj} x_j + b_k \right) \]

There are two basic approaches to the implementation of an analog artificial neural system: monolithic and modular. The first approach considers a monolithic block that is able to map the inputs to the output, as shown in Figure 3.

![Figure 2. Artificial Neuron](image2)

![Figure 3. Monolithic approach (“Black Box”)](image3)

The design of the “black box” is rather difficult due to the complex non-linear multi-input single-output mapping that the ANN controller needs to implement. On the other hand, the modular approach considers the architectural aspects of the ANN. Each artificial neuron is implemented and then interconnected with the others to build the ANN circuit.

In the modular approach, the architecture is kept unchanged so that the analysis and understanding of the circuit is made easier. Each neuron circuit is a building
block that can be designed as two different circuits: a body circuit and an activation function circuit (Figure 4). The former implements the synaptic weights and the summation block, whereas the latter, as the name implies, implements the activation function.

![Figure 4. Modular approach to neuron design](image)

For example, a circuit that implements a given activation function is a DC mapper that has voltage levels as its input and output. Figure 5 illustrates the operation of an activation function circuit. In the modular approach, many neuron circuits are needed for the implementation of a complete ANN controller.

![Figure 5. Activation function block](image)

Now the proposed evolutionary platform PAMA-NG is presented.

3. The EHW platform

The PAMA-NG (Programmable Analog Multiplexer Array – Next Generation) platform is based on the PAMA platform [13] that has been conceived as an environment for evolving analog circuits based on discrete components without the need of electronic circuit simulators.

The PAMA-NG is capable of evolving building blocks for intelligent analog systems [3][9] and is based on analog multiplexers/demultiplexers, as shown in Figure 6. These multiplexers are fixed elements and are responsible for the interconnections of the different discrete components that can be plugged into the board. The prototype platform can interconnect 32 component terminals (leads) to up to 16 circuit connecting points (circuit nodes) or analog channels [1][13]. Intrinsic evolution of analog circuits is achieved through a traditional Genetic Algorithm whose chromosomes are made up of genes (bit string). Each gene configures the select signals of a particular analog multiplexer.

![Figure 6. PAMA-NG block diagram](image)

The PAMA-NG block diagram shown in Figure 6 consists of three main blocks: Genetic Algorithm, I/O boards and Analog Reconfigurable Circuit. The I/O boards are connected to the PC bus responsible for A/D and D/A conversions and for chromosome download. Power supplies and signal conditioning circuits used to buffer the input and output signals are not shown in the diagram.

The digital output hardware is used for downloading the chromosome (128 bits) to the reconfigurable circuit. Some channels of the 32 channel 12 bit A/D converter hardware sample the inputs and output nodes; others can be used to monitor the circuits nodes and perform tasks such as checking voltage sources.

An eight channel 12-bit D/A converter allows the application of different input values to the neural circuits being evolved, so that the desired input to output mapping specified in a file (map.txt) may be performed.

The Analog Reconfigurable Circuit (ARC) is presented in the next section.
3.1. Analog reconfigurable circuit

The ARC of PAMA-NG can interconnect 32 component terminals (leads) to up to 16 circuit connecting points (circuit nodes) and is divided into three layers: discrete components, analog multiplexers and analog bus (Figure 6).

Each component terminal (lead) is plugged to a certain line of an analog bus (circuit node) through an analog multiplexer (16-to-1; 4 select bits). The chromosome bits (32 multiplexers x 4 select bits = 128 bits), which are sent by the PC, select for each analog multiplexer a specific node (line of the analog bus) to be connected to the component’s terminal. Each line of the analog bus corresponds to one interconnection point of the circuit; some of them can be associated with external points, such as the input signal, power supply, ground and circuit output, while others can correspond to internal points of interconnection of the circuit.

The ARC module implements a 32 16-to-1 analog multiplexer/demultiplexer (bi-directional) which allows the connection of up to 32 component’s leads to 16 circuit nodes. The low on-resistance ($R_{on}$) – approximately 100Ω – of the analog multiplexers prevents the reconfigurable circuit from damage during random configurations.

Another important feature of the platform is its flexibility with respect to granularity: its building blocks can be chosen by the user, from transistors to high level analog components, such as operational amplifiers.

3.2. Evolution of Artificial Neuron Circuits

The design begins by editing a special file named map.txt, which contains the desired input-output mapping that the circuit shall perform. Then, the designer chooses the discrete components, from low-level – transistors and resistors – to higher level circuits – such as operational amplifiers and comparators or even a more complex circuit. Evolution takes place after specification of the Genetic Algorithm (GA) parameters. Each chromosome is evaluated and the GA searches for the best individual (circuit topology) maximizing the fitness (between 0 and 1). The desired output is read from map.txt.

In the fitness technique used – exponential normalization – the probabilities of the ranked individuals being selected are exponentially weighted [15] and are given by:

$$p_i = \frac{c-1}{c^i-1} \frac{1}{N}$$

where $c$, which varies from 0 to 1, defines the exponential level of the method. The closer $c$ is to 1 the lower is the function’s exponential level, and, as a consequence, the selection intensity.

4. Experiments

The preliminary tests described here consisted of the design of a two-input one-output neuron circuit. The body circuit and the function activation circuit were evolved separately. Experiments with more complex neurons (more inputs and different activation functions) are still under way.

**Experiment 1**

This experiment focuses on the development of a body circuit with one bias and two inputs. The objective was to check the platform's capability of evolving a circuit that corresponds to the previously edited map.txt file.

The fitness function used to evolve the circuit was based on the error of the difference between the desired output, as specified in map.txt, and the actual output.

Recalling that the discrete electronic components can be chosen by the designer, two high level component – operational amplifiers – and some resistors have been
used in this experiment. This choice of components was based on known circuit configurations or on previous tests.

The circuit schematic of the evolved circuit is illustrated in Figure 7. The mux’s Ron resistance is shown as 0.1K resistors.

The GA parameters are:
- Generations: 200
- Population: 400
- Crossover: 0.7
- Mutation: 0.1
- Steady State: 50
- Exponential Normalization c parameter: 0.9

By applying the `map.txt` file input values to the circuit, it has been observed that the maximum error is about 4%. This result shows that the evolved circuit performs well within expectations.

**Experiment 2**

In this experiment an activation function block is evolved on the platform. Considering the non-linearity of the desired function, transistors were added to make the evolution easier, so that a different set of components was used: two operational amplifiers, two PNP transistors, two NPN transistors and resistors.

The original and actual activation functions illustrated in Figure 8 demonstrate the good performance of the evolved body circuit, where only a small mapping error takes place.

The GA parameters are:
- Generations: 300
- Population: 400
- Crossover: 0.65
- Mutation: 0.1
- Steady State: 50
- Exponential Normalization c parameter: 0.9

The circuit schematic diagram of the evolved circuit is illustrated in Figure 9. The multiplexer’s R_on resistance is represented as 0.1 KΩ resistors. The observed unusual transistor configurations occur quite often with EHW development tools and indicate their ability to search for different circuit topologies.

---

**Figure 7. Schematic of the best evolved neuron body circuit**

**Figure 8. Neuron activation function**

**Figure 9. Schematic diagram of the evolved circuit**
5. Conclusions

The design of artificial neurons for analog ANNs on a new FPAA development platform called PAMA-NG has been proposed and checked out. In the first test, both the body circuit and the activation function circuit of a simple neuron have been successfully evolved.

The prototype nature of the PAMA-NG does not allow for the implementation of a complete analog ANN controller; an integrated circuit solution is crucial for the implementation of a completely evolvable controller. Another drawback of the PAMA-NG is that the choice of the adequate set of components depends very much on the user. A better alternative would be to let the components be evolved during the process, possibly exploring the unique nature of CMOS technology [16].

One interesting feature of the platform is scalability. Ideally, several (integrated version) platforms would be connected together, each of them with a different target – automatically defined during evolution depending on the platform’s individual evolution process. Evolution would occur in parallel and the fitness value would be determined by taking into consideration the system’s final output and each platform’s individual output. Clearly, due to the analog nature of the platform, this scalability is somewhat limited. Nevertheless a complete analog solution to ANN controller remains appealing to certain niche applications.

Another interesting possibility would be the evolutionary design of ANN circuits on a commercial FPAA, e.g. those from Anadigm Corp.

6. References


