





---

# Evolution of Dynamical Neural Network Arrays to Correct Arrhythmias in a Simulated Human Heart

---

**John C. Gallagher**

Department of Computer Science and Engineering  
Wright State University, Dayton, OH 45435-0001

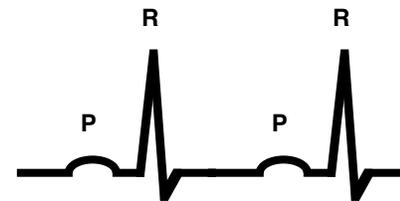
Email: jgallagh@cs.wright.edu    Phone: (937) 775-3929    Fax: (937) 775-5133

## Abstract

The study of evolutionary approaches to create electrical circuits is becoming increasingly widespread. This paper will present early results in evolving electrically implementable dynamical neural networks that correct several classes of arrhythmia in a simulated human heart. In addition to assessing the quality of the evolved circuits, we will examine how the evolved circuits differ from traditional pacemakers and consider what implications these differences may hold. We will also propose some objective function improvements that should increase the number of effective evolved controllers

## 1 INTRODUCTION

The author has recently proposed the use of Continuous Time Recurrent Neural Networks (CTRNNs) as an enabling paradigm for evolving analog electrical circuits [Gallagher, 2000] [Gallagher and Fiore, 2000]. In those previous works, we presented hardware implementations of extrinsically evolved CTRNNs and demonstrated that CTRNNs evolved in simulation are practically implementable and behave as predicted in simulation. Making the presumption of extrinsic evolvability, this paper will discuss an application of the CTRNN paradigm to the evolution of neural network based electrical circuits to correct common arrhythmias in a simulated human heart. We will begin with a brief discussion of cardiac arrhythmia as well as of the cardiac model employed. Second, we will describe the basic control strategy as well as the neural architectures and models utilized. Third, we will discuss the specific genetic algorithm (GA) methods used as well as the results of forty evolutionary searches aimed at producing arrhythmia correcting circuits. Fourth, we will qualitatively assess both evolutionary successes and failures with an eye toward improving GA search for future experiments. We will also provide a brief, qualitative description of the best controller's function. Finally, we will conclude with discussion of possible practical applications of these results and discuss future goals and research.

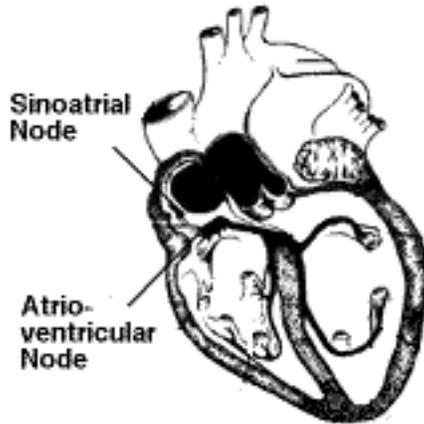


**Figure 1:** In an ECG recording, P waves correspond to atrial contractions. R waves correspond to ventricular contractions.

## 2 CARDIAC MODEL

The human heart has four chambers, two atria constitute the top half of the heart, and two ventricles constitute the heart's bottom half. In normal operation, the atria contract in unison followed shortly by the ventricles contracting in unison. The sinoatrial (SA) node (located in the upper portion of the right atrium -- see Figure 2) is autoexcitatory and serves as the heart's primary pacemaker -- firing an impulse capable of causing atrial contraction about sixty to seventy times per minute. The atrioventricular (AV) node (located near the interventricular septum near the bottom of the right atrium) is also composed of autoexcitatory tissue and fires with an intrinsic frequency of about 40 cycles per minute. The AV node is the only electrically conducting pathway between the atria and the ventricles. Atrial contractions (P waves) are initiated by autonomous firing of the SA node. The conventional view holds that those impulses are conducted through the AV node with a time delay -- triggering ventricular contraction soon after (R waves).

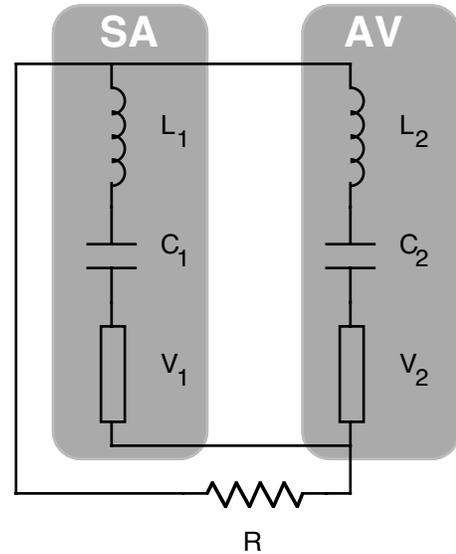
Cardiac arrhythmias are breakdowns in the normal relationships between atrial and ventricular contractions. One class of arrhythmias, the AV blocks, can be detected by examining electrocardiogram (ECG) recordings. ECGs are time series recordings of electrical activity in the heart. A simplified schematic representation of an ECG time series is shown in Figure 1. Three classes of AV blocks, character-



**Figure 2:** The positions of the sinoatrial and atrioventricular nodes in the human heart.

ized by specific ECG signatures, are commonly recognized. First degree AV blocks are characterized by a constant lengthening of the PR interval (the time between P and R waves) beyond normal length (greater than about 0.2 seconds in a human). Second degree blocks are characterized by the occasional failure of R waves to appear after P waves. There are two varieties of second degree blocks. In Mobitz Type I block (Wenckebach block), the interval between P and R waves continuously increases until eventually an R wave is completely dropped. In Mobitz Type II block, PR intervals are constant, but R waves are still intermittently dropped. Mobitz Type II blocks are further subcategorized into groups based on the ratio of atrial to ventricular contractions. A patient who drops every other R wave would be said to have 2:1 Mobitz Type II block, for example. Third degree blocks are characterized by total failure of synchronization between the top and bottom of the heart. In third degree block, the ventricles contract independently of the atria and at a rate near the AV node's intrinsic frequency of forty beats per minute.

For this work, we adopted a coupled oscillator model of heartbeat generation described in [Di Bernardo, Signorini, and Cerutti, 1998] and [Signorini, Cerutti, and Di Bernardo, 1998]. Both the AV and the SA nodes are represented by *van der Pol* oscillators (Figure 3). Each oscillator possesses an "active resistor" (labeled  $V_1$  and  $V_2$  in Figure 3) that is capable of both dissipating and producing energy. The value of the resistor  $R$  controls the degree of coupling between the oscillators. All three classes of AV blocks can be simulated by manipulating  $R$ . When  $R$  is zero, the oscillators are decoupled -- simulating 3rd degree block. At settings of  $R$  near 1.0, the oscillators are coupled and one observes normal 1:1 entrainment of the two halves of the heart with the SA node acting as the primary pacemaker. One observes other classes of AV block for intermediate values of  $R$ . The specific nature of the block observed depends additionally on the intrinsic frequencies of the SA and AV oscillators. This is



**Figure 3:** Schematic representation of coupled oscillator cardiac model. [Di Bernardo, Signorini, and Cerutti, 1998]

consistent with physiological observations. Patients are often "stress tested" to increase heart rate and uncover blocking phenomena that might otherwise go unnoticed.

We can express the state equations of the cardiac model as follows:

$$[1] \quad x_1' = \frac{1}{C_1} x_2$$

$$[2] \quad x_2' = -\frac{1}{L_1} [x_1 + g(x_2) + R(x_2 + x_4) + S_1]$$

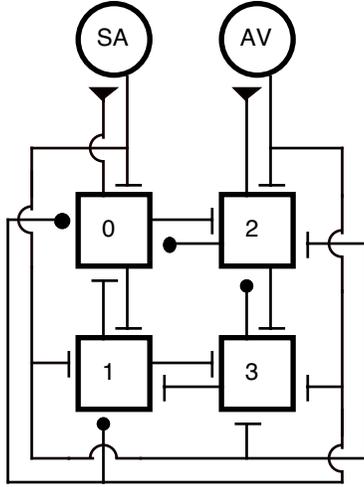
$$[3] \quad x_3' = \frac{1}{C_2} x_4$$

$$[4] \quad x_4' = -\frac{1}{L_2} [x_3 + f(x_4) + R(x_2 + x_4) + S_2]$$

$$[5] \quad f(x) = V_2(x) = -x + \frac{1}{3} x^3$$

$$[6] \quad g(x) = V_1(x) = -x + \frac{1}{3} x^3 + h(x)$$

$$[7] \quad h(x) = \begin{cases} -x^2 - 1/4 & |x| < 1/2 \\ -x & x > 1/2 \\ x & x < -1/2 \end{cases}$$



**Figure 4:** Generic CTRNA Architecture: The numbered squares are CTRNN neurons. Excitatory connections to neurons end in bars, inhibitory connections in filled dots. The circles at the top represent the SA and AV nodes in the cardiac model. Neurons 0 and 2 receive weighted input from the SA ( $x_2$ ) and AV ( $x_4$ ) nodes respectively. The outputs of neurons 0 and 2 are applied to the simulated heart as  $S_1$  and  $S_2$ .

In the model,  $x_2$  represents SA node activation and  $x_4$  represents AV node activation. Both correspond to voltages in the electrical equivalent model.  $x_1$  and  $x_3$  correspond to the electrical current through the SA and AV branches of the circuit respectively. The PR interval is taken to be the time between subsequent maxima of  $x_2$  and  $x_4$ .  $S_1$  and  $S_2$  are externally controlled voltage sources used to inject stimulation into the SA and AV nodes respectively. Settings of all model parameters are made to mimic physiological observations. Details can be found in [Di Bernardo, Signorini, and Cerutti, 1998]. We will discuss specific model settings used in this work in the sections of this paper in which they are relevant.

### 3 NEURAL MODEL

The evolved cardiac controllers are Continuous Time Recurrent Neural Arrays (CTRNAS) [Gallagher, 2000], which are grid arrangements of CTRNN neurons [Beer, 1995]. Each neuron in the grid receives inputs only from their immediate right, top, left, and bottom neighbors. These neighbors may be other neurons, sensory devices, or physical actuators. Each grid neuron has the following state equation:

$$[8] \quad \tau \frac{dy}{dt} = -y + I_r + I_t + I_l + I_b + I_s$$

$$[9] \quad output = \sigma(y + \theta)$$

$$[10] \quad I_r = right\_input \times r\_weight$$

$$[11] \quad I_t = top\_input \times t\_weight$$

$$[12] \quad I_l = left\_input \times l\_weight$$

$$[13] \quad I_b = bottom\_input \times b\_weight$$

$$[14] \quad I_s = output \times self\_weight$$

$$[15] \quad \sigma(x) = \left[ \frac{2}{1 + e^{-x}} \right] - 1$$

CTRNA controllers evolved in this paper were assembled using four neurons and interfaced to the cardiac model as shown in figure 4. Conventional artificial pacemakers operate by sensing P and R complexes and firing large depolarization spikes to either the top or bottom of the heart as appropriate when contractions are "missed". In a sense, they take large corrective action when a control algorithm determines that a necessary event did not occur. In this work, we're employing a different approach. Here we will attempt to evolve "neural prosthesis" that augment, rather than override, the natural cardiac conduction system. The CTRNAs are intended to apply small corrections continuously to prevent the heart from missing contractions to begin with, rather than force them to occur after they are missed. Since such subtle augmentation can be difficult, if not impossible, to design by hand, we turned to genetic algorithms to evolve neural array parameter settings that appropriately coupled our device to mend broken (albeit simulated) hearts.

### 4 GENETIC ALGORITHM SEARCHES

In our experiments, we assumed the AV node to possess an intrinsic frequency of 40 beats per minute (BPM). We also assumed the SA node to be the primary cardiac pacemaker and thus, modified its intrinsic frequency to drive the whole heart at faster or slower rates. All healthy simulated hearts, therefore, had the following parameter settings in common:

$$C_2 = 0.675 \quad L_2 = 0.027 \quad R = 1.1$$

The values of  $C_2$  and  $L_2$  were chosen to produce an intrinsic AV frequency of 40 BPM as well as to produce physiologically realistic AV action potential shapes. Details can be found in [Di Bernardo, Signorini, and Cerutti, 1998]. The setting of  $R$  was chosen to reflect normal, healthy coupling of the two oscillators.  $C_1$  and  $L_1$ , which specify the intrinsic frequency of the SA node, were chosen to produce rates of 40, 60, 80, 100, and 120 BPM in the whole heart when SA is normally coupled with the AV oscillator. SI settings that produce heart rates of interest are given in Table 1.

Table 1: Model Heart Parameter Settings

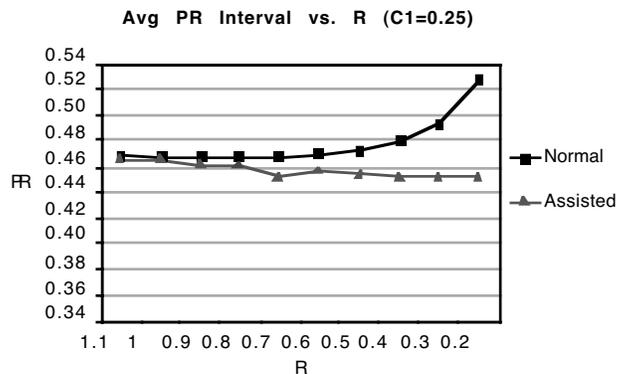
Heart Rate	$C_1$	$L_1$
40 BPM	0.395	0.079
60 BPM	0.250	0.05
80 BPM	0.15	0.03
100 BPM	0.144	0.0228
120 BPM	0.09125	0.01825

AV blocks were simulated by adjusting  $R$  to values in the range of [0..1.1] inclusive.

We used the public domain genetic algorithm package GAucsd<sup>1</sup> (version 1.4) to evolve all neural parameters of CTRNAs of the form shown in Figure 4. A neural array was specified by four time constants, four biases, and sixteen weights for a total of twenty-four parameters. Each grid parameter was encoded in four bits, with time constants  $\in [-0.5, 10]$ , and biases and weights in the range  $\pm 16$ . GAucsd employs a technique known as dynamic parameter encoding (DPE) that zooms the mapping between a fixed set of bits and a real parameter based on statistics gathered during a run [Schraudolph & Belew, 1992]. GAucsd parameters were set as follows: Total Trials = 100000; Population Size = 500; Crossover Rate = 0.62; Mutation Rate = 0.000160; Generation Gap = 1; Scaling Window = -1; Structures Saved = 10; Max Gens w/o Eval = 2; Options = Aclue; Maximum Bias = 0.99; Max Convergence = 112; Conv Threshold = 0.98; DPE Time Constant = 50; Sigma Scaling = 2.

Fitness of an individual CTRNA controller was evaluated by simulating the heart, coupled to the CTRNA as defined in Figure 4, for fifteen simulated seconds and assessing the errors between CTRNA augmented heart behavior and healthy, unaugmented (normal) heart behavior under ten pre-selected test conditions. The ten test conditions consisted of each of the five hearts defined in the earlier table (rates from 40 through 120 BPM) under normal ( $R=1.1$ ) and 3rd degree block ( $R=0.0$ ) conditions. It was assumed that any controller capable of correcting third degree block would also correct 2nd and 1st degree blocks. The total error for each test condition was the mean squared error (MSE) between the normal and augmented PR interval, the MSE between the normal and augmented RP interval, the MSE between normal and augmented atrial rate, and the MSE between the normal and augmented ventricular rate. The total error for a CTRNA controller was taken to be the sum of the errors of each of the ten test conditions.

Forty separate GA searches were run. Each required about two days to evaluate 200 generations on a 300 MHz



**Figure 5:** Average PR interval vs.  $R$  for unassisted and assisted heart beating at 60 BPM. Note progressive 1st degree block in the unassisted heart vs. largely constant PR intervals over a range of  $R$  values for the assisted heart.

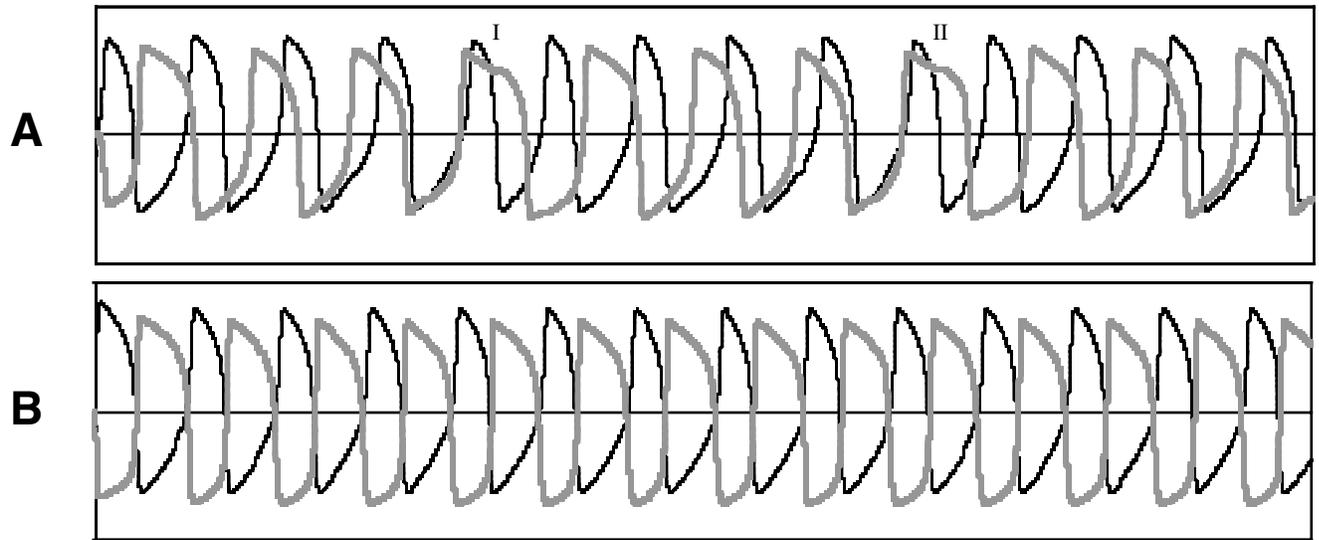
FreeBSD Pentium architecture workstation. Four searches resulted in CTRNAs capable of correcting all three classes of AV block in simulated hearts operating over the range of 40 to 120 BPM. We will call these devices "effective controllers". The remaining searches produced devices that could correct arrhythmias at slow heart rates, but failed in various ways to correct arrhythmias at faster pacings. We will refer to these devices as "ineffective controllers". First, we will examine the evolutionary histories and performances of the effective controllers, then discuss why the ineffective controllers never evolved properly.

## 5 ASSESMENT OF EFFECTIVE CONTROLLERS

Experiments are named after the workstation they ran on. We will use the experiment's name, as appropriate in context, to refer to both the GA search itself as well as the best CTRNA resulting from the run. Four GA experiments, *belle*, *spunky2*, *droopy2*, and *sprocket2* produced effective controllers at generations 52, 14, 41, and 14 respectively. Prior to the emergence of the effective controllers at the generations indicated, all four experiments showed a similar evolutionary history. The controllers first evolved the ability to correct all classes of AV block at 40 and 60 BPM, later evolving the ability to correct at 80 and 100 BPM, and finally, the ability to correct at 120 BPM. After the emergence of an effective controller, all searches produced only minor improvements to subsequent controllers by fine tuning PR and RP intervals.

Figure 5 demonstrates correction of 1st degree AV block by the CTRNA controller *belle*. This result is typical of the correction produced by all four effective controllers. Note that as  $R$  drops and coupling between the SA and AV nodes breaks down, the average PR interval increases in the unas-

<sup>1</sup> URL: <ftp://ftp.aic.nrl.navy.mil/galist/src/GAucsd14.sh.Z>



**Figure 6:** Time series plots for an unassisted heart with 2nd degree Mobitz type I block (A) and the same heart coupled with the evolved controller *droopy2*.  $x_2$  (SA activation) is shown using black thin lines.  $x_4$  (AV activation) is shown using bold gray lines. Atrial contractions occur at peaks of  $x_2$  and ventricular contractions occur at peaks of  $x_4$ . The PR interval is the length of time between  $x_2$  and subsequent  $x_4$  peaks.

sisted heart. Also note that the PR interval does not increase in the assisted heart. In fact, it decreases slightly.

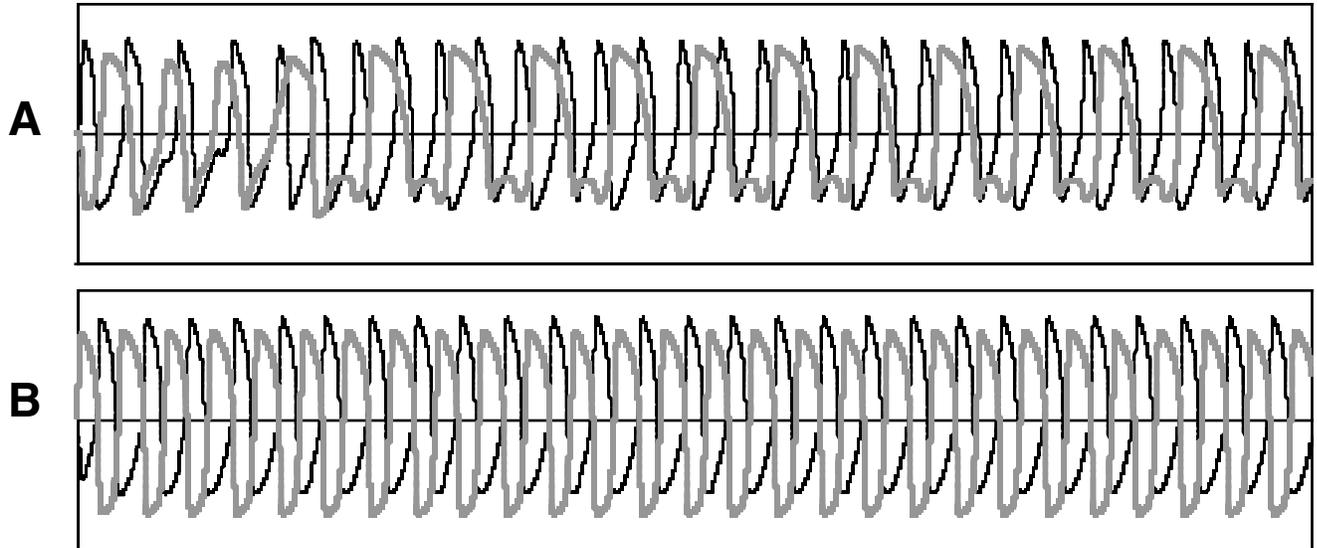
Figure 6 demonstrates typical correction of 2nd degree Mobitz type I block. Figure 6a shows time series plots of  $x_2$  (SA) and  $x_4$  (AV) activations over a period of fifteen seconds for a model heart beating at 60 BPM. Heart parameters are as indicated earlier in this section for a 60 BPM heart except that  $R=0.09$ . Notice the missed ventricular contractions at I and II in Figure 6a. Also note that the PR interval leading up to those missed beats continually lengthens leading up to the missed contraction. In Figure 6b we see  $x_2$  and  $x_4$  time series for the same heart coupled with the evolved controller *droopy2*. Note that in the CTRNA assisted heart, the PR intervals are constant at about 0.428 seconds. This compares favorably with the PR interval of 0.469 seconds in an uncorrected healthy heart ( $R=1.1$ ) beating at 60 BPM. The corrective ability shown for *droopy2* is typical of that provided by the other three successful controllers. This corrective ability also generalizes to Mobitz Type I block at other heart rates.

Figure 7 illustrates typical correction of 2nd degree Mobitz Type II 2:1 block. Figure 7a shows SA and AV activation time series for an unassisted damaged heart beating at 100 BPM. Heart parameter settings are as earlier indicated for a 100 BPM heart except that  $R = 0.26$ . Note that in Figure 7a PR intervals are constant and every other ventricular contraction is dropped. Figure 7b shows the time series of the same heart assisted by the evolved CTRNA *sprocket2*. Note that a 1:1 relationship between SA and AV activations has been

restored. Also note that the PR interval is constant. The corrective ability of *sprocket2* in this is typical of all four successful controllers. All four successful controllers have also been shown to be able to correct Mobitz Type II block at other heart rates and for other entrainment ratios.

Figure 8 demonstrates typical correction of 3rd degree AV block. Heart parameters are as indicated for a 120 BPM heart, except that  $R=0.0$  (total AV block). In Figure 8a, the time series of the unassisted heart, we observe total AV block. The SA and AV nodes are completely disassociated and AV fires at its intrinsic 40 Hz. Figure 8b shows the time series for the same heart assisted by the controller *spunky2*. After a short startup transient, the assisted heart beats at about 120 BPM. The startup transient is not typical. The other three successful controllers induce appropriate 120 BPM operation from time zero.

In addition to correcting all three classes of AV blocks tested at rates from 40 through 120 BPM, no successful controller interferes with proper operation of the simulated heart when the "natural" conduction system is intact ( $R=1.1$ ). Time series of healthy hearts with one of the four successful assist units engaged are nearly indistinguishable from time series of healthy hearts with the CTRNA assist units disengaged. When coupled to a "healthy heart" the four effective CTRNA controllers alter neither the normal SA paced heart rate nor the length of the PR interval.



**Figure 7:** Time series plots for an unassisted heart with 2nd degree Mobitz Type II block (A) and the same heart coupled with the evolved controller *sprocket2*. Plotting conventions are identical to those used in Figure 6.

## 6 ASSESMENT OF INEFFECTIVE CONTROLLERS

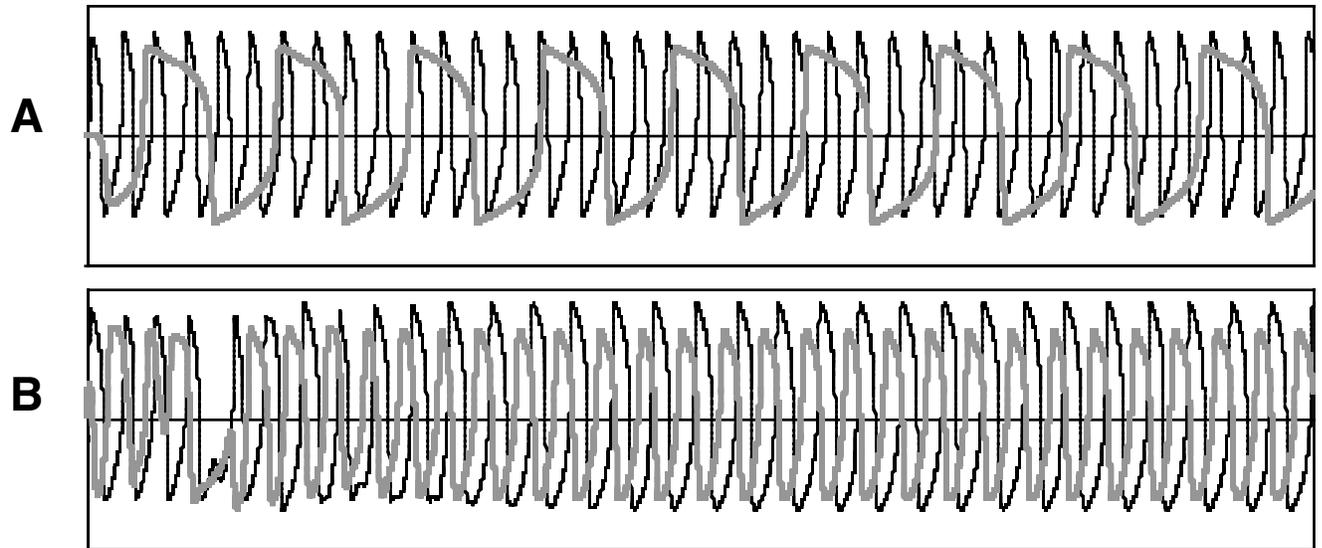
What of the 36 ineffective controllers? Though they do not correct for all classes of AV block over the whole range of heart rates studied, they are all effective in correcting for some subset of conditions and/or block types. Considering that many of these produced corrective action qualitatively similar to those provided by early stages of the four successful controllers and also considering that the two hundred generations allowed was nearly four times as many as needed by the slowest evolving effective controller -- one needs to ask why more experiments didn't result in successful controllers. The answer seems to lie in a poor choice of objective function. Though it is true that effective controllers score well on the stated objective function -- it is equally true that some ineffective controllers score as well or better. Use of average PR and RP intervals seems particularly problematic -- as it is possible for corrected heart rates to adopt non 1:1 SA to AV entrainments that produce average PR and RP intervals close to desired values. This can cause the GA to perpetuate clearly unacceptable solutions -- preventing convergence on good solutions. Many examples of exactly this phenomenon were found among the set of ineffective controllers. One possible fix, which is currently under study, is to modify the objective function to also minimize the variances of the PR and RP intervals as well as the error between assisted AV ratio and the perfect AV ratio of 1.0 (AV ratio is the ratio of atrial to ventricular contractions, or P to R waves). These modifications to our naive objective function should weed out unacceptable solu-

tions without significantly increasing the amount of computation time necessary to complete a GA search.

## 8 QUALITATIVE CIRCUIT ANALYSIS

Although a complete analysis of how the evolved CTRNA controllers function has yet to be completed, there are a few interesting functional observations that can be made at this time. For this discussion, we will focus our attention on the controller *belle*. Our qualitative observations of its function are general for all four effective controllers.

Refer to Figure 2 for a controller circuit diagram. In *belle*, neurons 1 and 3 maintain fixed outputs at all times. This means that we could remove those neurons from the system and replace them with constant biases to neurons 0 and 2. Dynamically speaking, neurons 1 and 3 are irrelevant. Neuron 3, which provides stimulation directly to the AV node, acts much like a traditional artificial ventricular pacemaker. It "listens" for SA activity and, with a slight time delay, conducts that activation to the AV node to trigger a ventricular contraction. Cutting the connection from neuron 2 to the AV node completely breaks the controller's ability to maintain 1:1 entrainment of the two halves of the heart. The activity of neuron 0 is somewhat more subtle and much more interesting. In our model, the SA node is the primary cardiac pacemaker -- its intrinsic frequency sets the pace for the whole heart. The heart, however, does not entrain to the exact SA frequency. The SA node is coupled to a "slow" 40 BPM oscillator, so at high target rates, the SA must adopt an intrinsic frequency somewhat above the target "whole heart" frequency. If we cut the connection from neuron 0 to



**Figure 8:** Time series plots for an unassisted heart with 3rd block (A) and the same heart coupled with the evolved controller *spunky2*. Plotting conventions are identical to those used in Figure 6.

the SA node, the simulated assisted heart beats at the SA intrinsic frequency, rather than the correct coupled frequency. At a target rate of 120 BPM, for example, the *belle* assisted heart with the neuron 0 to SA connection cut beats at 160 BPM. In practice, neuron 0 selectively slows the SA node. At whole heart target frequencies near 40 BPM, neuron 0 is almost totally quiescent. Since the SA and AV rates are both near 40 in that case, no damping is needed for the SA frequency. As the whole heart target frequency, and thus, SA intrinsic frequency, increase, neuron zero becomes progressively more active and progressively slows the SA node to its correct target value. Neuron 0, in a sense, acts as a rate sensitive damper of oscillations in the SA node. With traditional pacemaker therapy, such *tachycardia* (dangerously fast heart rate) would be treated with drugs to slow the heart rate. With our evolved circuit, the *tachycardia* induced by the breakdown of coupling between the heart halves is handled by the evolved assist device itself.

## 7 CONCLUSIONS AND FUTURE WORK

The author has proposed evolved CTRNAs as an enabling paradigm for compact, cheap, analog circuits for device control. This paper's basic contribution is an example of using that paradigm to create extremely compact controllers capable of correcting a large class of failure modes in a simulated heart. Although the yield of 10% (4 out of 40 controllers were totally effective) was disappointing -- we were able to identify the likely reason for that poor yield and are taking steps to fix the problem. Preliminary results are showing a near 100% yield of effective controllers using the improved

objective function. Further, the controllers from this new batch seem to operate like the four discussed in this paper.

More broadly, the work described here provides a base from which to spin off several potentially intriguing lines of inquiry. One avenue being explored is the evolution of CTRNAs that correct for a wider variety of heart defects. Of particular interest would be CTRNAs capable of correcting bradycardia (dangerously slow heart rates) and tachycardia (dangerously fast heart rates) in addition to arrhythmia. In preliminary work, the author has already evolved one four-neuron CTRNA that subsumes all the abilities of the best controllers introduced in this paper and can, in addition, automatically correct bradycardia. In that circuit, the unused neurons (neurons 1 and 3) oscillate, providing an internal clock that drives the heart if its intrinsic rate falls dangerously low. We've already observed some correction of tachycardia in circuits already evolved. This has led us to attempt evolving circuits that can slow one or both sides of the simulated heart as needed. Initial studies on this front are currently underway.

To be accepted as effective controllers, evolved systems must be understandable and explainable. This can be difficult for evolved electrical circuits in general and for evolved neural nets specifically. The author and his colleagues have enjoyed considerable success in using dynamical systems principles to explain the principles underlying evolved CTRNN central pattern generators [Beer, Chiel, and Gallagher, 1999][Chiel, Gallagher, and Beer, 1999]. The controllers evolved in this work are much more intricately coupled to sensory input than any device we have previously considered and will provide an excellent opportunity to ex-

tend our CTRNA analysis techniques. Such study is being actively pursued.

Although control of the simulated heart was initially conceived as a pure research problem against which to test the effectiveness of evolved hardware to control an interesting dynamical system, the very fact that the interesting dynamical system is a cardiac model raises the question of evolving CTRNA devices to assist real hearts. The model used in this work reproduces many important aspects of cardiac function -- but in its simplicity falls short in properly representing a variety of physiological details. Although useful in helping us develop GA and CTRNN analysis methods, the simple coupled oscillator model would be of limited value in predicting actual behavior when coupling an assist device to a real heart. The author is investigating more physiologically accurate models against which to test the evolved controllers. The author is also investigating more accurate modeling of the interface between the CTRNA and heart tissue. Actual implantable voltage sources used to produce cardiac stimulation are significantly less perfect than the ideal sources we presumed in this paper. Likewise, actual cardiac sensor leads may not provide clean signals -- nor do they have the same properties over the lifetime of the device. Appropriate modeling of sensory and excitation leads are both currently under investigation.

In conclusion, this paper has presented an example of evolving several practically implementable neural network based circuits that can effectively control a dynamically rich system, preventing a range of possible failure modes. As a pure research problem, it provides both an interesting GA benchmark and a source of interesting analog neural networks upon which to develop new dynamical systems analysis techniques. In addition, this work sets the stage for potential applied research in neural prosthesis and bio-control.

## Acknowledgments

The author would like to thank Steven Perretta and Wendy Peluso for their comments and suggestions on this manuscript.

## References

- Beer, R.D. (1995). On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior* 3(4):469-509.
- Beer, R.D., Chiel, H.J. & Gallagher, J.C. (1999). Evolution and analysis of model CPGs for walking II. General principles and individual variability. *J. Computational Neuroscience* 7(2):119-147. (Kluwer).
- Chiel, H.J., Beer, R.D., & Gallagher, J.C. (1999). Evolution and analysis of model CPGs for walking I.

Dynamical modules. *J. Computational Neuroscience* 7(2):99-118.

- Di Bernardo, D.D., Signorini, M.G., and Cerutti, S. (1998). A model of two nonlinear coupled oscillators for the study of heartbeat dynamics. *International Journal of Bifurcation and Chaos* Vol. 8, No. 10
- Gallagher, J.C. and Fiore, J.M. (2000). Continuous time recurrent neural networks: a paradigm for evolvable analog controller circuits. in *The Proceedings of the National Aerospace and Electronics Conference*. IEEE Press.
- Schraudolph, N.N. & Belew, R.K. (1992). Dynamic parameter encoding for genetic algorithms. *Machine Learning* 9(1).
- Signorini, M.G., Cerutti, S. and Di Bernardo, D.D. (1998). Simulation of heartbeat dynamics: a nonlinear model. *International Journal of Bifurcation and Chaos*. Vol 8., No. 8

---

# Evolutionary Mechanisms for Smart On-board Adaptive Sensing applied to the MECA Electrometer

---

Didier Keymeulen, Adrian Stoica, Martin Buehler, Ricardo Zebulum and Vu Duong

Jet Propulsion Laboratory  
California Institute of Technology  
4800 Oak Grove Drive  
Pasadena, CA 91109  
818-354-4280  
didier.keymeulen@jpl.nasa.gov

## Abstract

In-situ exploration as required for example by missions to comets and planets with unknown environmental conditions, has recently been approached with new ideas, such as the use of biology-inspired mechanisms for hardware sensor adaptation. The application of evolution-inspired formalisms to hardware design and self-configuration lead to the concept of evolvable hardware (EHW). EHW refers to self-reconfiguration of electronic hardware by evolutionary/genetic reconfiguration mechanisms. In this paper we describe the initial development of efficient mechanisms for smart on-board adaptive sensing, adaptively controlling the reconfigurable pre-processing analog electronics using evolvable hardware, which will lead to higher quality, lean data

## 1 INTRODUCTION

High data rates provided by modern sensors surpass on-board real-time processing capabilities. This is addressed by imposing large on-board storage memory and high communications bandwidth; there is however no good solution to using the data in real-time control situations such as fast entry, descent and landing, or within sensor webs. Only a small fraction of the data carries quality information, yet current pre-processing electronics is not smart enough to eliminate useless/redundant data. In fact more information could be obtained from the sensor if the electronics would adapt to incoming signals and the context of the measurement.

The concept of reconfigurable and adaptive electronics for signal conditioning has led to a series of recent chips that allow programmable adjustment of amplifier gains, memory-based compensation of sensor nonlinearity, etc [17]. However, the flexibility of these programmable devices is limited by the high level of reconfiguration

granularity, and require that all compensation data is predetermined through lab experiments and then stored in ROM; also no later changes in sensor characteristics or electronics itself could be considered once the sensor is in operation.

A complementary technique, called evolvable hardware (EHW), allows the automatic determination of optimal electronic circuit configurations[1][2][3]. In particular a chip designed for evolvable hardware experiments, the Jet Propulsion Laboratory (JPL) Field Programmable Transistor Array (FPTA) has high flexibility by reconfiguration at transistor level [20]. Evolutionary algorithms allow for automatic determination of optimal configuration. In the narrow sense EHW refers to self-reconfiguration of electronic hardware by evolutionary/genetic reconfiguration mechanisms as in our application [4][5][6][9]. In a broader sense EHW refers to various forms of hardware, from antennas to complete evolvable space systems that could adapt to changing experimental environments and, moreover, increase their performance during the mission.

In this paper we describe the initial development of efficient mechanisms for smart on-board adaptive sensing, adaptively controlling the reconfigurable pre-processing analog electronics using evolvable hardware, which will lead to higher quality, lean data. The target is to demonstrate the mechanisms on an adaptive electrometer providing the same or more information content than the MARS'01 MECA (Mars Environmental Compatibility Assessment) Electrometer with a significant reduction in the total amount of transmitted data. The electrometer was part of MECA project and has as objective of the project to gain a better understanding of the hazards related to the human exploration of Mars.

In the paper we identify one application of adaptive sensor array device for which the reduction of the data can be considerable: discrimination task of materials with different triboelectric properties. The discrimination task requires a sophisticated signal conditioning able to analyse multiple responses in order to extract differences

in signal and adapt to deal with the high sensitivity of the sensor array to ambient conditions. The analysis and sensitivity are translated to requirements and fitness evaluation metrics that are used by an evolutionary algorithm to determine the optimal adaptation mechanisms.

This paper reports on experiments that illustrate how evolutionary algorithms can design analog circuit integrated in the sensing elements and adapted to the experimental conditions. At this stage of the research, the search for an electronic circuit realization of a desired transfer characteristic is made in software as in *extrinsic* evolution. In *extrinsic* evolution, the final solution is downloaded to (or becomes a blueprint for) the hardware. In the near future we will use *intrinsic* evolution where the hardware actively participates in the circuit evolutionary process and is the support on which candidate solutions are evaluated.

This paper is organized as follows: Section 2 presents a description of the electrometer sensor array. Section 3 presents the adaptive sensor architecture. Section 4 presents an evolution-oriented architecture for reconfigurable hardware based on the concept of FPTA and the details of the evolutionary design tool. Section 5 presents the experiments and results obtained for the adaptive electrometer for a discrimination application in a changing environment. Section 6 concludes the paper.

## 2 ELECTROMETER SENSOR ARRAY

The electrometer is part of MECA project. The objective was to gain a better understanding of the hazards related to the human exploration of Mars [15][16]. The MECA project also has a material patch experiment to determine the effects of dust adhesion, a wet chemistry laboratory with ion selective electrodes to characterize the ionic content of the soil, and microscopy station with optical and atomic force microscopes to determine particle size and hardness.

The electrometer was built into the heel of the Mars '01 robot arm scoop as seen in Fig. 1. The instrument has four sensor types: (a) triboelectric field, (b) electric-field, (c) ion current, (d) temperature. The triboelectric field sensor array contains five insulating materials to determine material charging effects as the scoop is dragged through the Martian regolith. The insulating materials were chosen after Earth-based tests in Mars simulant soils.

During digging operation the electrometer is out of the way. After digging, the scoop is rotated so the electrometer head is pointing down toward the Martian soils allowing it to be rubbed against the Martian soil.

In the rubbing sequence, depicted in Fig. 2, the scoop is first lowered against the Martian soil. During the start of the traverse, the electrometer is zeroed by closing a switch which will be discussed later. After reaching the end of its traverse, the scoop is abruptly removed from the soil at which time the triboelectric sensor response is measured.

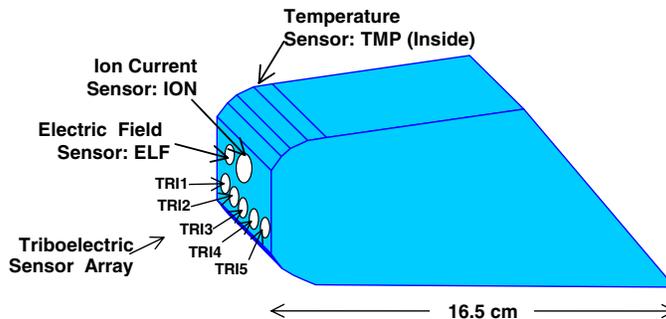


Figure 1: Electrometer sensor suite mounted in the heel of the Mars'01 scoop

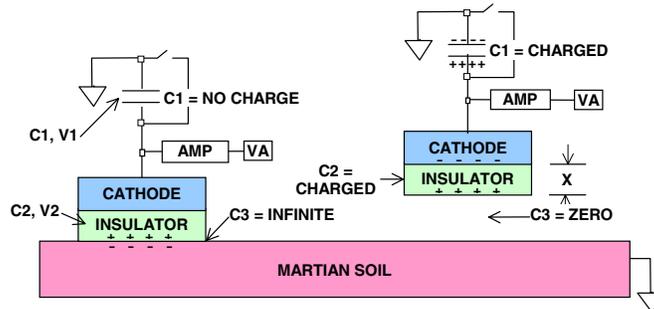


Figure 2: Operational scenario for the scoop and charge distribution in the electrometer during rubbing (left) and after removal from the surface (right).

As seen on the left in Fig. 2, charge is generated triboelectrically across capacitor C3 as the insulator is rubbed on the Martian surface. Since the charges are in close proximity across C3, no charge appears across capacitors C1 or C2. As the insulator is removed from the surface, the charges redistribute themselves across C1 and C2 according to the charge relationship  $Q1 = Q2$  and provide the signal for the amplifier.

This electrometer is an induction field meter [11] operated in a direct current mode, where the operational amplifier input current charges C1. The electrical schematic of the non-adaptive component of the triboelectric sensors is shown in Fig. 3. The design of the electric field sensor follows from the traditional electrometer [12]. The instrument is composed of a capacitive divider where C2 is the field sensing capacitor and C1 is the reference capacitor. The point between the capacitors is connected to the positive terminal of the first stage amplifier (terminal +5 of U3) operated in the follower mode. The sensing electrode is protected by a driven guard that is connected to the negative terminal of the first stage amplifier (terminal -6 of U3). A second operational amplifier (U4) is added to provide additional amplification. At the beginning of the measurements, C1 is discharged using the solid-state switch, S1 which has very low leakage. In the TRI sensor, C2 has an insulator dielectric which acquires charge during rubbing.

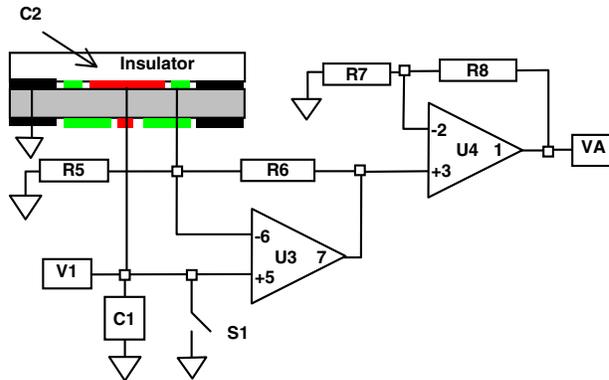


Figure 3: Schematic circuit representations for the non-adaptive component of the Triboelectric sensor (TRI) fully characterized before field use.

Four different insulating materials were loaded into the titanium triboelectric sensor head. A typical experiment consists of manually rubbing a wool felt on the triboelectric head at room temperature. The results are shown in Figure 4. The falling period between 10 and 20 seconds represents the rubbing period. The large negative response is for the Rulon-J which is to be expected for Rulon-J rubbed on wool.

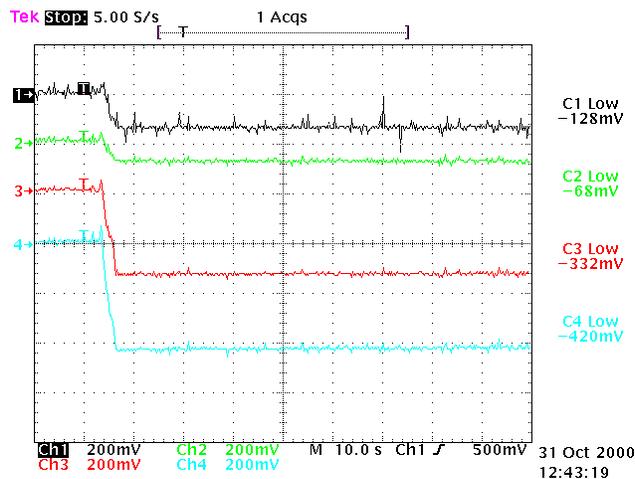


Figure 4: Response of triboelectric sensor array to white wool felt (For all figures: response C1 is ABS (TRI1), response C2 is Polycarbonate (TRI2), response C3 is Teflon (TRI3) and response C4 is Rulon-J (TRI4)).

### 3 ADAPTIVE SENSOR ARCHITECTURE

The triboelectric sensor array is an example of a hybrid integrated array devices where the sensors are grouped on the same devices but where the signal processing is done on a separate device as we will describe later [13]. This sensor array employs similar sensors (in terms of the measurand) but sensors have subtle differences (i.e. partially correlated outputs) related to the triboelectric properties of materials, known as the triboelectric series. The triboelectric sensors have poor specificity and so respond to a very wide range of materials. The signal

processing must then carry out a sophisticated analysis of the responses to extract the subtle differences in signals. The approach we have chosen, as shown in Fig. 5, is to use an evolvable hardware discriminator signal conditioner connected to the triboelectric sensor array and that will be able after evolution to discriminate with high precision the response of different materials.

Another important reason to use an adaptation mechanism is to be able to do in-situ self-calibration [14]. Indeed the sensors are very sensitive to ambient conditions, such as temperature, humidity, atmospheric and contact pressure, ambient gas, materials. They are also sensitive to the material and surface condition of the sensors. For example the dust cling on the insulator surface affect considerably the response of the triboelectric sensor arrays. Finally the array sensor has poor ageing characteristic, that is the triboelectric sensing element is slowly corroded and thus changes its response characteristics with time. To remedy this high sensitivity to the ambient conditions and sensors conditions, we performed an in-situ self-calibration: calibrate the sensors right at site with the current environmental conditions and a set of given sensor materials.

Fig. 5 shows the basic arrangement of an adaptive electrometer array sensor system for discriminating different materials. The triboelectric property of the material is sensed by an array of sensors, each with its response which is converted to an electrical signal via suitable transduction circuitry. The voltage signal  $VA_i$  is then injected to the evolvable hardware specially designed for the current environment and a set of materials. The prediction of the triboelectric property of the material is given in terms of voltage. In the next section, we describe the evolvable hardware developed by JPL, called FPTA and the mechanism to find the best circuit configuration to perform the classification task.

### 4 EVOLUTION-ORIENTED DEVICES AND ENVIRONMENT

The idea of a FPTA was introduced first by Stoica in [8]. The FPTA is a concept design for hardware reconfigurable at the transistor level. As both analog and digital CMOS circuits ultimately rely on functions implemented with transistors, the FPTA is a versatile platform for the synthesis of both analog and digital (and mixed-signal) circuits. Further, it is considered a more suitable platform for synthesis of analog circuitry than existing FPGAs or FPAAs, extending the work on evolving simulated circuits to evolving analog circuits directly on the chip.

The FPTA module is an array of transistors interconnected by programmable switches. 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 a 1 represents an ON switch and a 0 represents a

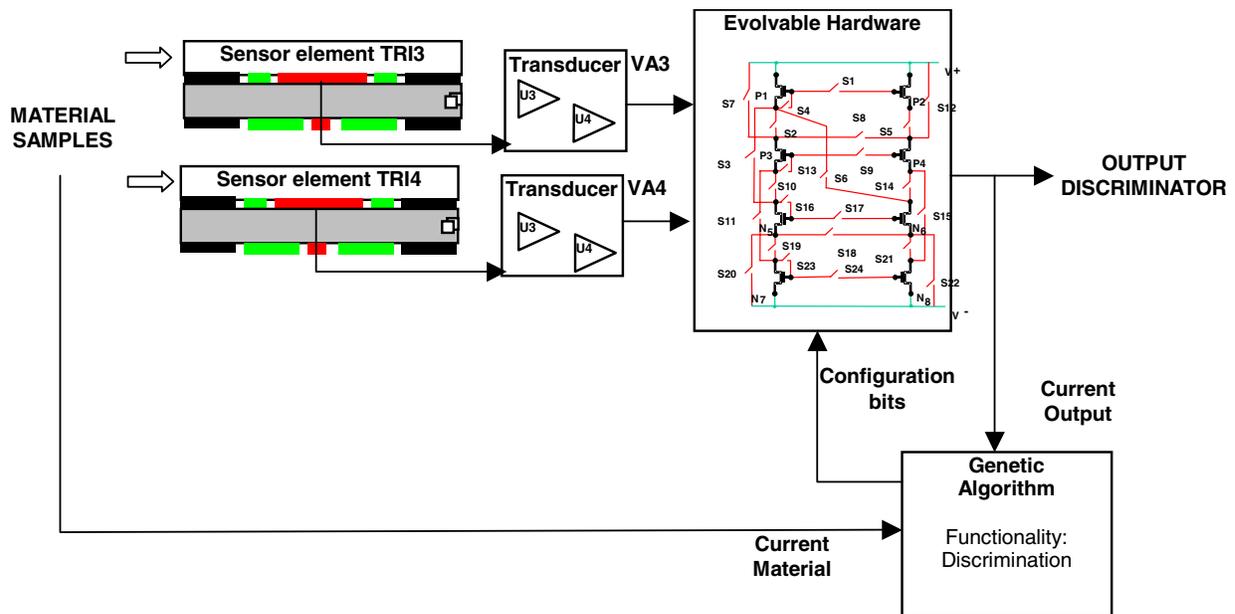


Figure 5. Schematic arrangement of an adaptive electrometer sensor array device

OFF switch. The FPTA architecture allows the implementation of bigger circuits by cascading FPTA modules with external wires. To offer sufficient flexibility the module has all transistor terminals connected via switches to external terminals (except for power and ground). Issues related to chip expandability were treated in elsewhere [10].

Figure 6 illustrates an example of a FPTA module consisting of 8 transistors and 24 programmable switches. In this example the transistors P1-P4 are PMOS and N5-N8 are NMOS, and the switch-based connections are in sufficient number to allow a majority of meaningful topologies for the given transistor arrangement, and yet less than the total number of possible connections. Programming the ON and OFF switches defines a circuit. The effects of non-zero, finite impedance of the switches can be neglected in the first approximation. One FPTA module was fabricated as a Tiny Chip through MOSIS, using 0.5- $\mu\text{m}$  CMOS technology. We build a testbed for future development with a test board with four chips mounted on it and connected with the electrometer (Fig. 7).

In the context of electronic synthesis on reconfigurable devices such as the FPTA, the architectural configurations are encoded in "chromosomes" that define the state of the switches connecting elements in the reconfigurable hardware. The main steps in evolutionary synthesis of electronic circuits are illustrated in Figure 8. First, a population of chromosomes is randomly generated to represent a pool of circuit architectures. The chromosomes are converted into control bit strings, which are downloaded onto the programmable hardware. In the particular case of the FPTA cell, the chromosome has 24 bits that determines the state of the 24 switches (Figure 6). Circuit responses are compared against specifications of a target response using as fitness the root mean square error.

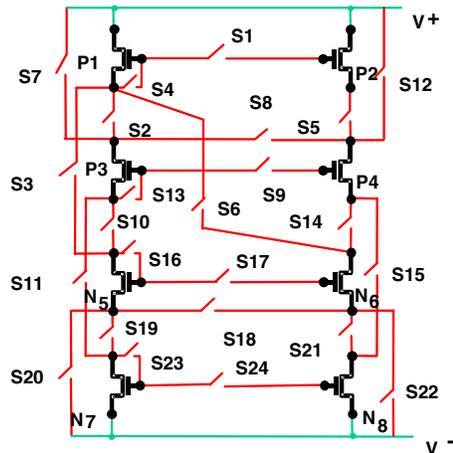


Figure 6: Module of the Programmable Transistor Array

The individuals are ranked based on their fitness; that is, how close they come to satisfying the target. Preparation for a new iteration loop involves generation of a new population of chromosomes from the pool of the best individuals in the previous generation. Individuals are selected probabilistically based on their fitness. Some are taken as they were and some are modified by genetic operators, such as chromosome crossover and mutation. The process is repeated for a number of generations, resulting in individuals with increasingly better fitness. The genetic algorithm is usually ended after a given number of generations, or when the closeness to the target response has been reached. In practice, one or several solutions may be found among the individuals of the last generation.

In addition to the procedure described above, which is called *intrinsic Evolvable Hardware* or *hardware evolution*, Figure 8 also shows an alternative way to carry on evolutionary circuit synthesis, by the use of simulators instead of reconfigurable chips. In this particular case, the chromosome is mapped into a SPICE circuit model, which is simulated and evaluated. This later procedure is called *extrinsic Evolvable Hardware* or *software evolution*. The mapping of the chromosome into the circuit netlist is accomplished by examining the chromosome values bit by bit. According to each bit value (0 or 1), the state of its corresponding switch will be set in the circuit netlist. After all the switches' states are determined, the circuit is simulated. The extrinsic approach has been used for the experiments of the adaptive electrometer sensor array. The intrinsic approach is currently under development.

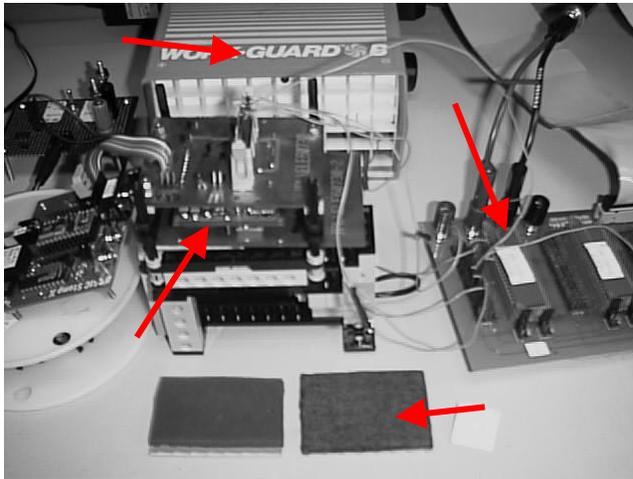


Figure 7. Module of the Programmable Transistor Array connected to the electrometer

An evolutionary design tool EHWPack (Figure 8) was developed to facilitate experiments in hardware and software evolution [7], as defined in the previous section. This tool incorporates the public domain Parallel Genetic Algorithm package PGAPack as genetic engine running on a UNIX workstation. Referring to software evolution, we incorporated in the EHWPack, the SPICE 3F5 as circuit simulator. In the case of hardware evolution, the tool proved very useful in testing architectures of reconfigurable hardware and demonstrating evolution on FPTA reconfigurable chips. An interface code links the GA with the hardware where potential designs are evaluated, while a Graphical User Interface (GUI) allows easy problem formulation and visualization of results. At each generation the GA produces a new population of binary chromosomes, which get converted into configuration bits downloaded into the 4 FPTAs reconfigurable chips or into Netlists that describe candidate circuit designs, and are further simulated by SPICE.

## 5 ADAPTIVE SENSOR EXPERIMENT

One experiment was conducted. The experiment shows that the evolvable hardware approach finds a FPTA circuit that is able to discriminate between the responses of the electrometer to three different materials.

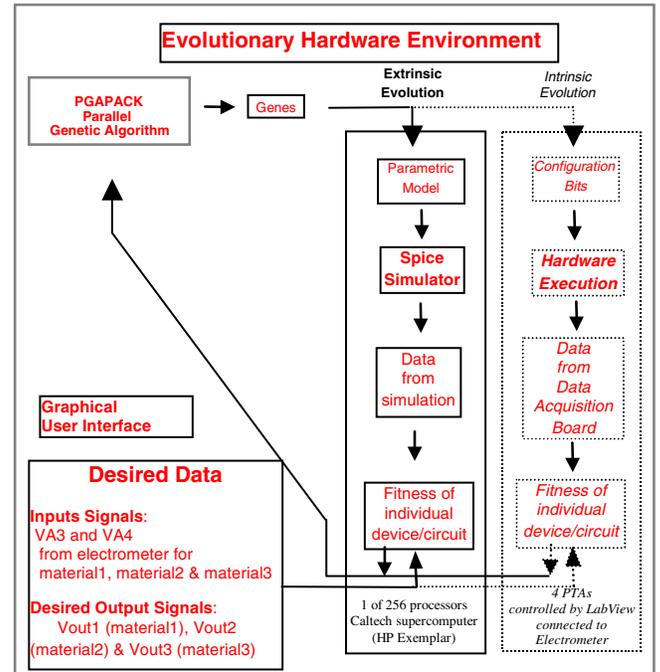


Figure 8. Environment for evolutionary design

The experiments used three rubbing material samples (wool felt, Teflon and Polystyrene) and used only two insulating materials of the electrometer (Teflon and Rulon-J). The experiments start by an initialization procedure which puts the electrometer in a known state: the five electrometer insulators were cleaned by brushing followed by Am-241 alpha particle deionization. The deionization process was observed by running a trace and noting when the response no longer changed. After cleaning and deionization, the samples were placed in the apparatus as seen in Figure 7. The data acquisition was started and five points were acquired every second. The first fifty points are baseline points. During the next 200 points, the samples were rubbed by the apparatus from left to right as shown in Fig. 4, Fig. 9 and Fig. 10. During the final data points, the rubbing was stopped and the rubbing material was no longer in contact with the electrometer insulating materials.

At this stage of the research, the response of the electrometer to three materials was obtained by rubbing the materials on the electrometer. The resulting data was used to find a circuit able to discriminate between the response of the electrometer to the different materials by *extrinsic* evolution using the SPICE simulator.

The evolutionary experiment was conducted in air at a pressure of 970mb, relative humidity of 33 percent and a temperature of 21°C. The evolvable hardware system

used one FPTA cell. The circuit had two inputs and one output. At the two inputs, we injected the sensor responses of the insulating material TRI3 (Teflon, response C3) and TRI4 (Rulon-J, response C4) to the three rubbing materials in addition to the baseline as shown in Fig.4, 9 and 10. The outputs are collected as a voltage signal.

different materials are used for rubbing. It can be described by the following equation:

$$Fitness = \frac{1}{T} \int \sum_{i \neq j} |V_i(t) - V_j(t)|$$

where the indexes  $i$  and  $j$  sweep the four patterns of the three materials and the baseline and  $T$  is the period of time used to evaluate the fitness.

The main task of evolution is to synthesize a circuit able to discriminate among the three materials and the baseline by amplifying the voltage differences among the materials measured by the sensors. Figure 12 depicts the evolved circuit.

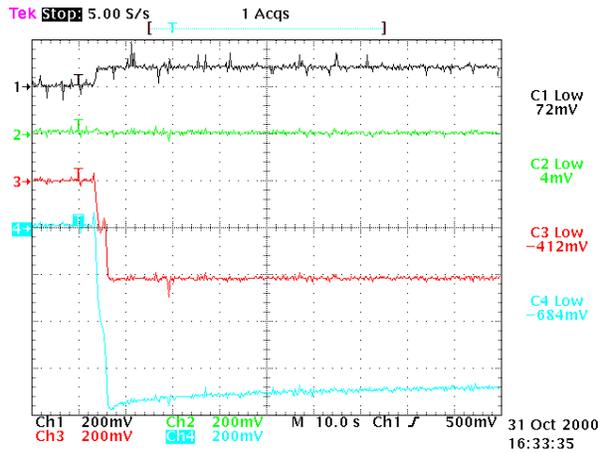


Figure 9. Response of triboelectric sensor array to **Polystyrene** (C1 is ABS, C2 is Polycarbonate, C3 is Teflon, C4 is Rulon-J). The four material samples are rubbed after 15[s].

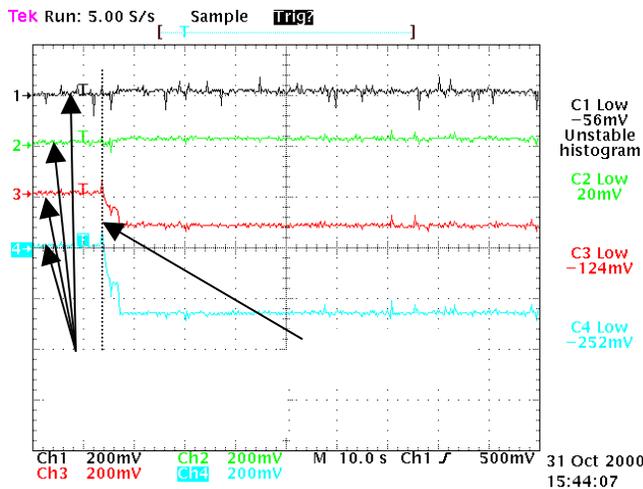


Figure 10. Response of triboelectric sensor array to **Teflon** (C1 is ABS, C2 is Polycarbonate, C3 is Teflon, C4 is Rulon-J).

The following GA parameters were used: Population: 40, Chromosome size: 24 bits for 1 FPTA, Mutation rate: 10%, Crossover rate: 90%, exponential Selection, Elite Strategy: 20% population size. The fitness function seeks to maximize the voltage difference at the output when

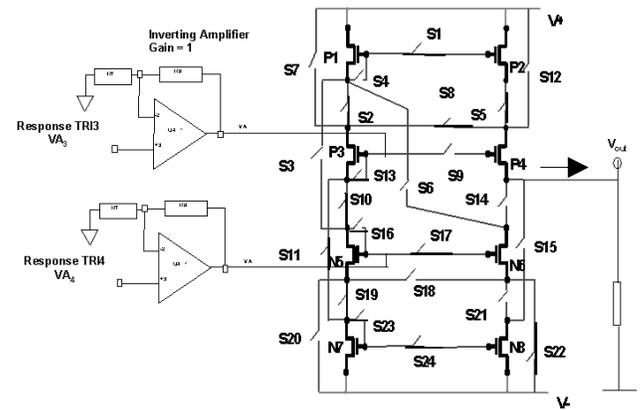


Figure 12. Evolved circuit able to discriminate among 3 materials and 1 baseline.

Figure 13 shows the response of the evolved circuit. In the negative part of the graph are the responses of the electrometer to the 3 materials and the baselines. Before being applied to the FPTA, they pass through a unit gain inverter stage (Fig. 12). In the positive part of the vertical axis, the circuit response for the four patterns is shown. In the circuit response, there is an average separation of 0.6V between the adjacent materials, except for the wool felt and teflon materials, for which the difference is 1.2V. The overall output range achieved a value around 2.3V, whereas the input range given by the responses of the sensor is around 0.7V. We observe also that the gain of the evolved FPTA is not constant: it depends on the amplitude of the input signal in such a way that the circuit improved the discrimination margin for different materials (Table 1).

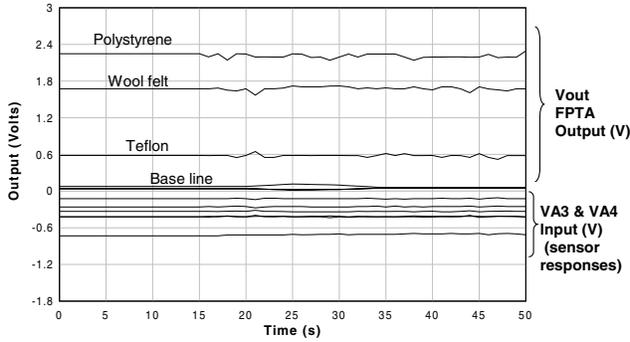


Figure 13. Response of the evolved circuit for 3 materials and 1 baseline. The time starts when the material sample is rubbed on the isolating materials of the electrometer

Table 1: Variable Gain of the Evolved Circuit

	Teflon	Wool	Polystyrene
VA3 (TR13)	0.124 V	0.332 V	0.412 V
VA4 (TR14)	0.252 V	0.420 V	0.684 V
(VA3+VA4)/2	0.188 V	0.376 V	0.548 V
Vout (output FPTA)	2.3 V	1.7 V	0.5 V
<b>GAIN</b>	<b>12.23</b>	<b>4.5</b>	<b>0.9</b>

To assess the generalisation of the circuit solution we have tested the evolved circuit with sensor responses with slightly different environmental conditions which resulted in a decrease in the response of the sensors. As expected, the difference in response of the evolved circuit was smaller but it still captured the correct order of the patterns corresponding to the triboelectric series [18,19] (Figure 14).

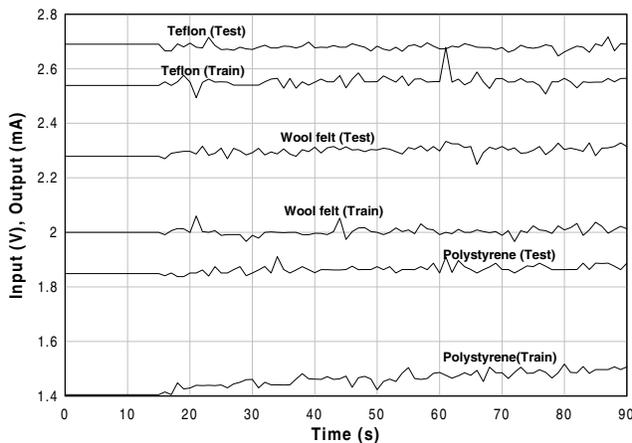


Figure 14. Response of the evolved circuit for 3 materials for slightly different environmental conditions than for experiment of Figure 13. The output measures the output current  $I_{out}$  at the drain of transistor P4.

## 6 CONCLUSIONS

These initial experiments, while illustrating the power of evolutionary algorithms to design analog circuit for sophisticated analysis of responses of sensor array and to maintain functionality by adapting to changing environments, only prepare the ground for further questions. Examples of further questions include addressing how the evolutionary mechanism can be implemented in hardware such that it can be integrated in the sensor, or how should the sensors responses be stored to avoid repeating the experiments for evaluating each circuit configuration.

The long term results of the proposed research would allow sensor electronics to adapt to incoming data and extract higher quality data, making available information otherwise not accessible. It will make sensor systems adaptive and intelligent. It will increase the amount of information available from sensors, while actually decreasing the amount of data needed for downlink.

### Acknowledgments

The research 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 Defense Advanced Research Projects Agency (DARPA) under the Adaptive Computing Systems Program and the National Aeronautics and Space Administration.

### References

- [1] J. Lohn, J. and S. Colombano, “Automated Analog Circuit Synthesis using a linear representation”, M. Sipper, D. Mange and A. Perez-Urbe (Eds.) *Evolvable Systems: From Biology to Hardware*, Springer-Verlag Lecture Notes in Computer Science Berlin 1998, pp. 125-133
- [2] A. Thompson, “An evolved circuit, intrinsic in silicon, entwined in physics”. In *International Conference on Evolvable Systems*. Springer-Verlag Lecture Notes in Computer Science, 1996, pp. 390-405.
- [3] E. Sanchez and M. Tomassini (Eds.) *Towards Evolvable Hardware*, LNCS 1062, Springer-Verlag, 1996
- [4] T. Higuchi, M. Iwata, and W. Liu (Eds.) *Evolvable Systems: From Biology To Hardware*, *Proc. of the First International Conference, ICES 96*, Tsukuba, Japan, Springer-Verlag Lecture Notes in Computer Science, 1997.
- [5] M. Sipper, D. Mange, A. Perez-Urbe (Eds.) *Evolvable Systems: From Biology To Hardware*, *Proc. of the Second International Conference, ICES 98*, Lausanne, Switzerland, Springer-Verlag Lecture Notes in Computer Science, 1998.
- [6] J. R. Koza, F. H. Bennett III., D. Andre and M. A. Keane, *Genetic Programming III – Darwinian Invention*

and Problem Solving, Morgan Kaufman, San Francisco, 1999

[7] Stoica, A. Klimeck, G. Salazar-Lazaro, C. Keymeulen, D. and Thakoor, A. Evolutionary design of electronic devices and circuits, *Proc. of the 1999 Congress on Evolutionary Computation*, Washington, DC, July 6-9, 1999

[8] Stoica, A. Toward evolvable hardware chips: experiments with a programmable transistor array. *Proceedings of 7<sup>th</sup> International Conference on Microelectronics for Neural, Fuzzy and Bio-Inspired Systems*, Granada, Spain, April 7-9, IEEE Comp Sci. Press, 1999.

[9] P. Marchal et al. Embryological development on silicon. In R. Brooks and P. Maes, editors, *Artificial Life IV*, pages 365-366. MIT Press, 1994.

[10] Stoica A., Keymeulen D., Tavel R., Salazar-Lazaro C., Li W. In *Proceedings of the First NASA/DoD Workshop on Evolvable Hardware*, pages 76-84. IEEE Computer Society Press, 1999.

[11] J. A. Cross, *Electrostatics: Principles, Problems and Applications*, Adam Hilger (Bristol, UK).

[12] "Electrometer Measurements", Keithley Instruments (Cleveland OH), 1972

[13] Julian W. Gardner, *Microsensors: Principles and Applications*. New York: John Wiley & Sons, 1994.

[14] Gert van der Horn and Johan L. Huijsing, *Integrated Smart Sensors: Design and Calibration*. New York: Kluwer Academic Publishers, 1998.

[15] M. Buehler, *Rocking Mineral Experiments*, Internal Technical Report, JPL. 16 January 2000. Pasadena, CA.

[16] M. Buehler et al., MECA Electrometer: Initial Calibration Experiments. In *Proceedings of the 10<sup>th</sup> International Conference (Electrostatics 1999)*. Institute of physics Conference Series No 163, pp 189-196, Institute of Physics Publishing, Bristol, UK, 1999.

[17] B. Schweber, Programmable analog Ics: designer's delight or dilemma ?. In *EDN*. April 2000. Cahners Publication.

[18] O.J. McAteer, *Electrostatic Discharge Control*. McGraw-Hill, 1989.

[19] N. Jonassen, *Electrostatics*, Chapman & Hall, 1998.

[20] A. Stoica et al.. Reconfigurable VLSI Architectures for Evolvable Hardware: from Experimental Field Programmable Transistor Arrays to Evolution-Oriented Chips. In *IEEE Transactions on VLSI*. April 2001, IEEE Press.















