2015 IEEE Symposium Series on Computational Intelligence

An Investigation of Underlying Physical Properties Exploited by Evolution in Nanotubes Materials

Stefano Nichele, Odd Rune Lykkebø, and Gunnar Tufte
Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
Email: {nichele, lykkebo, gunnart} @ idi.ntnu.no

***Abstract*—Computational materials, e.g. single-wall carbon nanotubes and polymer nanocomposites, have been evolved to solve complex computational problems. Such blobs of material have been treated as a black box, e.g. some input is encoded, some configuration signals are evolved to "program" the material machine, and some output is decoded. However, how the computation is performed, i.e. which physical properties are exploited by evolution to solve a given computational task, is not well understood. The general idea is that some undelying physical properties of the chosen material are exploited, e.g. capacitance, resistance, voltage potential, signal frequency, etc. In this paper we investigate which practical strategies are exploited by evolution on a simple (non-abstract) task: maximize or minimize amplitudes of output signals when square waves are used as input. This allows identifying an evolvability range for materials with different physical characteristics, e.g. nanotubes concentration. Inspection of evolved solutions shows that the strategies used by evolution to exploit physical properties are often unanticipated. This work is done within the European Project NASCENCE.**

I. Introduction

Evolution-in-Materio (EIM) [17], [16] is a relatively new field of research that explores new physical materials to perform computation. Such emergent computation is exploited by manipulating the chosen material via computer controlled evolution (CCE) [7], [8]. CCE may program the materials with different kinds of stimuli, e.g. voltages, currents, temperature, and magnetic fields. The undelying principle is that materials may intrinsically possess some physical mechanism that may compute. In contrast to a traditional design process where a computational substrate, e.g. silicon, is precisely engineered, EIM uses a bottom-up approach to manipulate materials. Different material substrates, e.g. liquid crystals, carbon nanotubes, field programmable gate arrays, have been succesfully used to solve computational tasks of different complexities (more details in the Background section). In all such cases, materials were treated as "black box", i.e. interfaced to a traditional computer; input signals were encoded and output signals decoded. Evolutionary algorithms have been used to search for suitable configuration signals to "program" the material as to be able to carry a wanted computation. How is this computation performed? At which physical level? Which intrinsic physical properties of the material allow computation to take place? At the current stage of research, all those questions are still unanswered. One motivation is that often the nature of the investigated problem abstracts the input and the output from the underlying physics, i.e. the fitness function is problem dependent and detached from the real physics of the used material substrate. As such, it is very difficult to know which range of outputs can be evolved for given inputs and configuration signals on a chosen material. We show later in the Result section that often the exploited physical properties are not intuitive.

In order to be able to pinpoint which physical properties are exploited by artifical evolution to produce a fitness increase, we define the problem of maximizing or minimizing the difference of output amplitudes on two different output pins. This may allow to evolve very similar outputs (if the material underlying physical properties allow so) or as different as possible, being able to identify a range of evolvability for different material samples. The analysis of different evolved solutions may highlight which strategies are utilized to solve the described task, as fitness is not abstracted but it is directly derived from the raw physical output, i.e. a purely electric response.

The article is laid out as follows: Section II gives background information on Evolution-in-Materio. In Section III motivation for this work is given and current issues with carbon nanotube materials are outlined. Section IV describes the expetimental setup and methodology. In Section V the results are presented together with discussion and analysis. Finally, Section VI concludes the article and gives directions for future work.

II. Background

Pask pioneered EIM in the 1950s without computers [3]. Using electric current he achieved self-assembly of neural structures in ferrous sulphate solutions [22]. The structure of the wires and the behaviour could be changed through external influence. Later, Thompson [24] demonstrated that evolution could be used to exploit physical properties of Field Programmable Gate Arrays (FPGAs) to solve computational tasks. Thompson found that it was impossible to replicate the evolved chip behavior in a simulation because evolution exploited undelying physical properties of the material. Harding and Miller [6] used liquid crystals displays as computational substrate. In the EU project NASCENCE [2], novel nano-scale materials, e.g. carbon nanotubes / polymer composites, nanoparticles, graphene, are exploited and configured to produce computation. In particular, carbon nanotubes have been

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.175

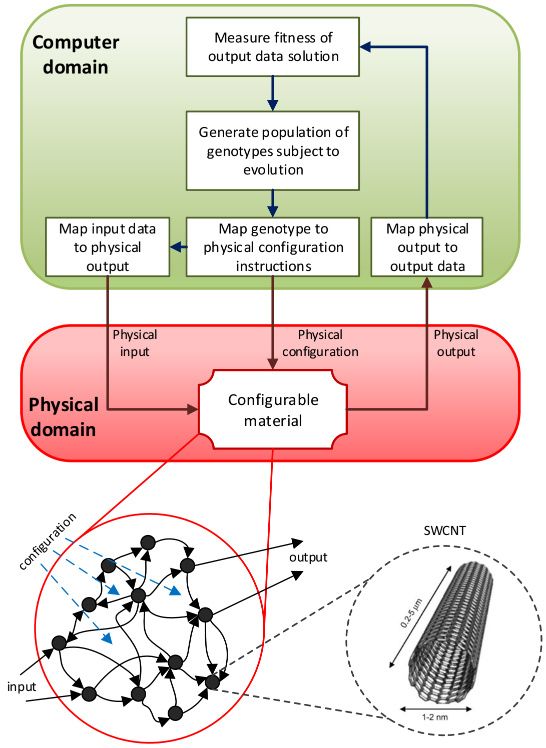

Fig. 1: Overview of Evolution-in-Materio: in the computer domain an evolutionary algorith generates configuration instruction to program the material. Those are tested against given inputs and physical outputs are mapped back and evaluated for fitness. Adapted from [4].

shown promising for the solution of Travelling Salesman [4], logic gates [10], and function optimization problems [19]. Input and configuration signals of different kinds have been used, e.g. static voltages [4], square wave voltages [12], a mix of both [14]. Square waves of different frequencies demonstrated potential to achieve a computationally rich behaviour [21] on single-walled carbon nanotubes / polymer nanocomposite materials (as the one used in the work herein). As such, they are the main subject of investigation in this paper. Figure 1 shows an overview of EIM where a physical material is interfaced to a computer running an evolutionary algorithm.

III. Motivation

At the current state of research, there are several practical issues for configuring physical materials for computation. One of the main problems is related to repeatability of results. Given a set of inputs and a set of configuration parameters, the resulting output may be unstable. This leads to devices or solutions that may work only once due to some imprecisions in the utilized input and measuring equipment, e.g. signals scheduling or noise, or due to changes in the material substrate. Such changes may be related to physical differences in the material, e.g. liquid crystals orientation for LCD based materials, or electrical changes for material that hold capacitance and can store charge. Practically, the latter can be reduced by allowing some transient period for the material to relax and reset to its original state or by applying a set of randomized inputs and configurations to avoid memory effects. Repeatability and stability issues lead to evolvability problems, as typically solutions are found by computer controlled evolution (CCE).

From an evolutionary point of view, having a working solution at a given generation that does behave differently in a subsequent generation means that evolutionary search is moving in the fitness landscape from a different point than expected, as the material physical state has changed. If we imagine the material as a circuit of interconnected nanocomponents, the fitness evaluation may change the values of the circuit components or the circuit topology, thus reapplying the same input configuration may lead to different performances. Such substrate changes may be considered as part of the genotype-phenotype mapping. Unfortunately, there is no guarantee that all the molecules in the material are in the same exact configuration as before. This can be problematic for evolvability. It was shown in [5] that it may be possible to evolve a new working solution if evolution is initialized with a previously working solution on the used material. In that case, evolution would be able to rapidly converge again on another similar (yet working) solution.

Another aspect that should be considered is the range of evolvability of different material samples. Intuitively, physical materials may have different characteristic that restrict or delimitate the range of evolvable solutions. Some of the parameters that may have an impact on evolvability and computational power are: intrinsic, e.g. internal physical properties of the molecules that compose the material (type, composition, electrical properties as conductivity or charge), external/environmental, e.g. external stimuli that influence temporarily or permanently the material properties (current, temperature, light), and construction, e.g. decided when the material is built (concentration of molecules, electrodes materials, size, pitch). For more details on this see [21].

The nature of several computational problems requires more than a single output, e.g. TSP in [4] requires 9 to 11 outputs, classification in [20] requires two outputs, robot controller in [18] requires also two outputs. Let us consider a problem solved in-materio where two output values are required (for example the problem in [18], where a robot controller is evolved in-materio for controlling the speed of motor wheels to navigate a maze) and assume two different materials are tested: the first one with similar electrodes coverage and a second where one of the two output electrodes is barely covered by conducting material. It is evident that the range of evolvable values for the first material is likely to be reasonably equal and the output mapping/encoding may be the same for both output values. On the other hand, with the second material, there are physical impediments for the less covered electrode to evolve the same range of outputs as in the better conducting electrode. As such, evolution may be able to overcome this issue by a different evolutionary strategy or, more likely, this may act as negative factor for evolvability. It may be that evolution discovers a strategy where the robot moves always to the right and the output value on the electrode connected to the right wheel is always higher, or it may be a physical limitation of the material (electrode coverage) and different output decoding may do the trick.

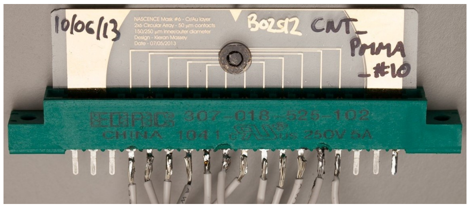

Fig. 2: Example of glass slide with microelectrode array covered by carbon nanotube / polymer nanocomposite

For those reasons, it is of high importance investigating which underlying physical properties are exploited by evolution to find solutions, as a 'black box' approach may have the described limitations.

IV. Setup and Methodology

In order to investigate what is the range of expected outputs in computational materials, or what is the atteinable output difference, a problem with evolved fitness that is related to measurable physical quantities is identified. The goal is to evolve the maximum or minimum raw output voltage value given a fixed number of configuration/input signals where square wave frequencies are applied. Square waves of different frequencies demontrated potential to achieve a computationally rich behavior [21], [13]. With such an approach, we may be able to identify how evolution exploits physical properties to give meaningful output, e.g. higher fitness in case of evolved solutions. Fitness is often disconnected from real physics, i.e. abstract measure suitable for the given task. Here the fitness is related to a measurable physical quantity: voltage output difference on different output pins.

Two different material samples are investigated: one with low nanotubes concentration (0.53% by weight) and one with high nanotubes concentration (5.00% by weight). Both slides provide 16 electrode contacts within the material and were fabricated by mixing single-walled carbon nanotubes (SWCNT) and poly(butyl methacrylate) (PMMA) dissolved in anisole (methoxybenzene). SWCNT are conducting or semi-conducting while PMMA creates uneven insulating regions within the nanotube network. Materials with higher SWCNT concentration act more as a conductive layer while lower SWCNT concentration creates more uneven distribution of nanotubes and polymer molecules, thus allowing non-linear current vs. voltage characteristics, as long as the network percolation threshold is reached. The material samples are interfaced to a computer running the genetic algorithm through a custom-built HW board called Mecobo [12]. The Mecobo board was built within the EU project NASCENCE and is used as interface between the microelectrode array slide hosting the material and the computer executing the evolutionary algorithm. All the input signals and output measurements are carried out through the Mecobo board. Mecobo offers the possibility of mixed signals, i.e. digital and analogue, input/output set-up. In the experiments herein the inputs, i.e. stimuli, are constrained to the digital domain but the material outputs, i.e.responses, are analogue. However, as in all set-ups including digital processing, the response is sampled. Thereby Analogue to Digital Converters (DACs) are in the signal chain. The Mecobo board offers AC signals and a possibility to connect (or disconnect) any input or output to any electrode. The connection/disconnection is implemented as bidirectional tristate ports. In the experiments presented, a differential output voltage is the goal. No absolute zero or reference level is thereby necessary. As such, there are no constrains set for evolution not to exploit the dynamic range of the ADC, i.e AC signals. In the results this possibility is clearly visible. The evolved solutions exploit the possibility of placing the signal in a favourable range, e.g. Figure 30, by exploiting the tristate ports and the available analogue voltage range. For more details on the typical setup see [12], [14], [13]. An example of material slide is shown in Figure 2.

A fixed number of input pins is used, starting from 1 and up to 10. The number of output pins is set to 2, as to be able to measure any output difference. The selection of input and output pins is under evolutionary control. Input signals consist of digital square waves with frequencies in the range 400 Hz - 25000 Hz, and duty cycle in the range 0% - 100%. Both frequency and duty cycle are under evolutionary control. The amplitude is fixed to 3.3 V, from 0 V to 3.3 V. Each input signal is applied simultaneously for 25 ms. The output signals are sampled at 250000 Hz for 5 ms, between the 10th and the 15th ms of computation, producing roughly 1200 data samples for each output pin. The output values represent the resulting voltage potentials on the output electrodes. The output fitness is represented in Equation 1, where the sum of the absolute values of the voltage differences is calculated.

$$Fitness = \sum_{i=0}^{len(out\_buf)} |Out1_i - Out2_i| \tag{1}$$

Two sets of experiments are executed, one where the goal is to maximize the fitness function (maximum output difference), and one where the goal is to minimize it (minimum output difference). Each experiment is executed for 100 generations and repeated 10 times. The used evolutionary algorithm is a $1 + \lambda - ES$, with $\lambda = 4$. In such scheme, the population size is $1+\lambda$ and the genotype with best fitness is selected as parent for the new population. The other individuals of the population are generated by mutating the parent. If there are no offspring with better fitness than the parent, but at least one has the same fitness as the parent, the offspring becomes the new parent. Figure 3 shows the genotype mapping. Each of the offspring undergoes a single mutation, i.e. one gene is mutated. If a gene representing an input pin is mutated, either the pin number, the frequency or the duty cycle is changed. If an output pin is mutated, the output pin number is changed. Mutation of frequency or duty cycle is performed by replacing the old value with a new random value in the correct range. Mutation of pin number is performed by swapping the selected pin (either input or output) with one of the free pins. Note that free pins are not under evolutionary control; they are represented here for convenience as to be able to perform a swap mutation with input/output pins more easily.

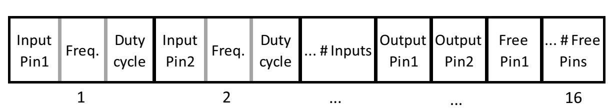

Fig. 3: The evolvable genotype representing input pins, output pins and free pins for a total of 16 genes.

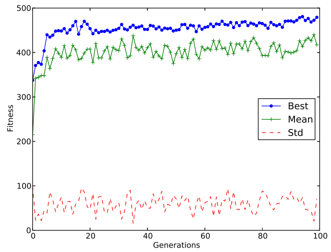

Fig. 4: 5,00% nanotubes, evolve maximum difference on 2 output pins, 1 single input.

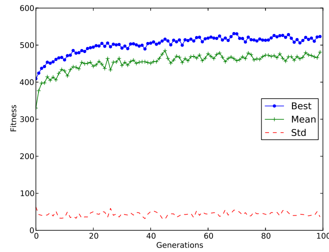

Fig. 5: 5,00% nanotubes, evolve maximum difference on 2 output pins, 5 input pins.

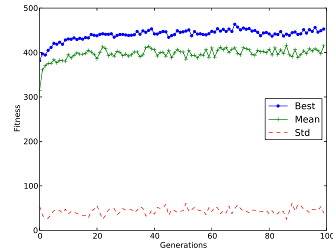

Fig. 6: 5,00% nanotubes, evolve maximum difference on 2 output pins, 10 input pins.

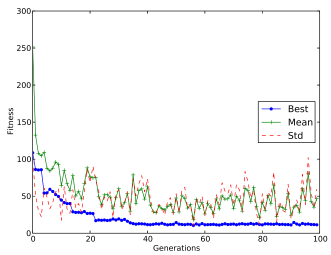

Fig. 7: 5,00% nanotubes, evolve minimum difference on 2 output pins, 1 single input.

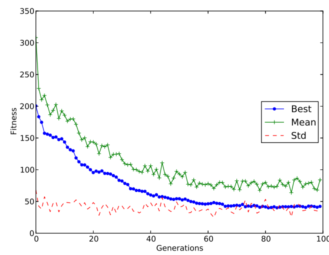

Fig. 8: 5,00% nanotubes, evolve minimum difference on 2 output pins, 5 input pins.

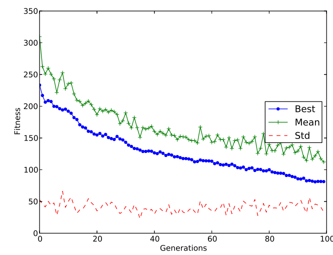

Fig. 9: 5,00% nanotubes, evolve minimum difference on 2 output pins, 10 input pins.

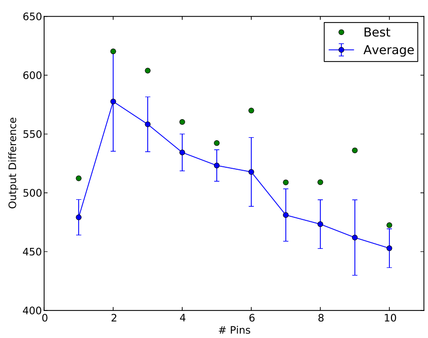

Fig. 10: 5,00% nanotubes, evolve maximum difference on 2 output pins, summary.

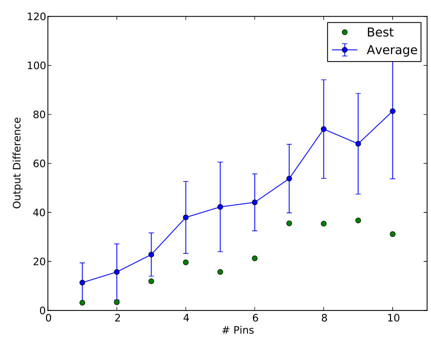

Fig. 11: 5,00% nanotubes, evolve minimum difference on 2 output pins, summary.

V. Results and Discussion

A. Evolve Maximum and Minimum Output Difference

This section presents the results for evolution of maximum output difference with an increasing number of input pins (Figures 4, 5, 6 give examples for 1, 5 and 10 input pins), followed by evolution of minimum output difference (Figures 7, 8, 9) tested on the high SWCNT concentration sample (5.00%). A summary is provided in Figure 10 (maximization) and Figure 11 (minimization). The same tests are repeated for the low SWCNT concentration sample (0.53%). Again example plots for maximization are given for 1, 5 and 10 inputs in Figures 12, 13, 14 and for minimization in Figures 15, 16, 17, respectively. A summary is provided in Figure 18 (maximization) and Figure 19 (minimization). A comparison of the *"ranges of evolvability"* is given in Figure 20 and Figure 21 for both samples.

For the material with high SWCNT percentage, it is possible to notice that evolution is slow and hardly manages to achieve significant fitness inprovements. This is visible in Figures 4, 5, 6, where output difference is maximized. The plots show a fairly steady situation after the first generations and increasing the number of inputs does not provide any benefit. This is in line with our hypothesis that higher concentration of conductive elements in the material makes it behave as a more uniform conductive layer where output differences are hardly evolvable. Having more than 2 input pins does not seem to help evolving higher output difference, as depicted in Figure 10. On the other hand, minimizing the difference seems a more evolvable task for the considered sample, as

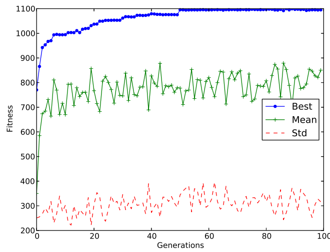

Fig. 12: 0,53% nanotubes, evolve maximum difference on 2 output pins, 1 single input.

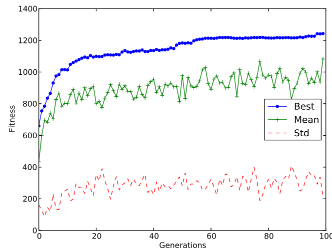

Fig. 13: 0,53% nanotubes, evolve maximum difference on 2 output pins, 5 input pins.

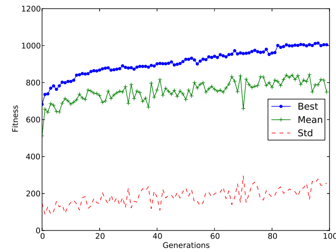

Fig. 14: 0,53% nanotubes, evolve maximum difference on 2 output pins, 10 input pins.

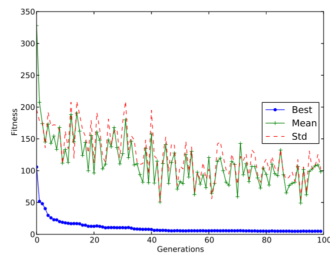

Fig. 15: 0,53% nanotubes, evolve minimum difference on 2 output pins, 1 single input.

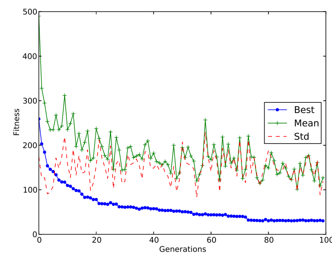

Fig. 16: 0,53% nanotubes, evolve minimum difference on 2 output pins, 5 input pins.

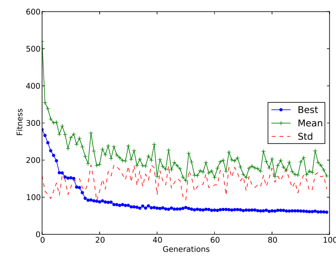

Fig. 17: 0,53% nanotubes, evolve minimum difference on 2 output pins, 10 input pins.

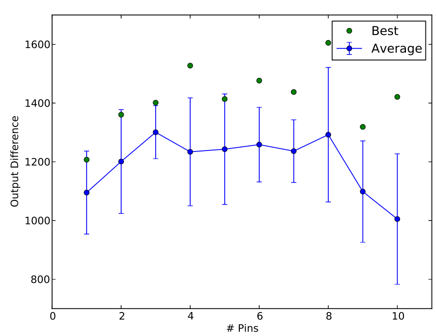

Fig. 18: 0,53% nanotubes, evolve maximum difference on 2 output pins, summary.

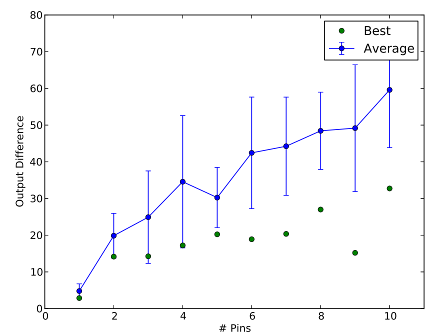

Fig. 19: 0,53% nanotubes, evolve minimum difference on 2 output pins, summary.

shown in Figures 7, 8, 9. As expected, increasing the number of input pins makes the task harder and the resulting minimum difference increases (within the given number of generations), as shown in Figure 11 . Note that for all the evolutionary runs, the standard deviation is plotted with error bars.

For the material with lower SWCNT percentage, things are different. Intuitively, it is easier to evolve greater output difference, as shown in the numerical results in Figures 12, 13, 14, but surprisingly adding more input frequencies does not improve evolvability, as visible in Figure 18 where there is no significant difference between 2 and 8 inputs. It was expected that evolving minimum difference would be more difficult on such sample. This is confirmed by results in Figures 15, 16, 17. Also in this case, adding more input signals makes it harder to minimize output difference. This is shown in Figure 19.

Overall, the ranges of evolvable differences on the different samples are depicted in Figure 20 and Figure 21. It is evident that the choice of the material sample has a very high impact on the evolved output signals. The evolvability range for the lower concentration sample is on average more than double than for the high concentration sample. For example, with two input pins, the sample with 5% concentration produced results in the range 42-577 and the sample with 0.53% concentration in the range 176-1201.

In order for any kind of computation to take place in a material, one of the key requirements is Ashby's law of requisite variety [1], which states that "in order to deal with the diversity of problems, a (control) system needs to have a repertoire of responses which is at least as many as those of the problem". Ashby's law underlines the importance of the

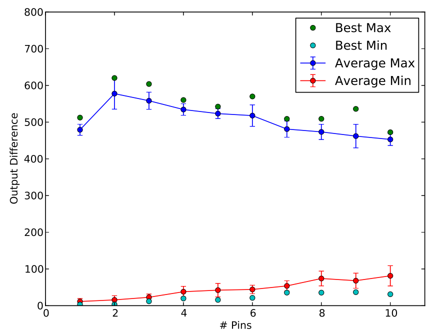

Fig. 20: 5,00% nanotubes, comparison summary, evolvability range.

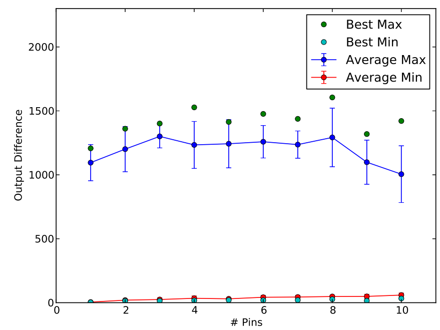

Fig. 21: 0,53% nanotubes, comparison summary, evolvability range.

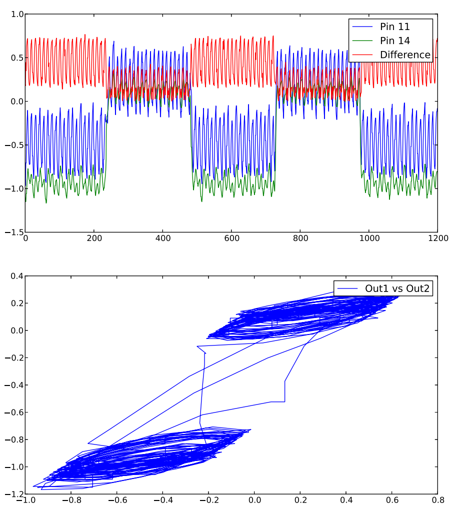

Fig. 22: Material B09S12, 0.53% nanotubes, example with 4 input pins.

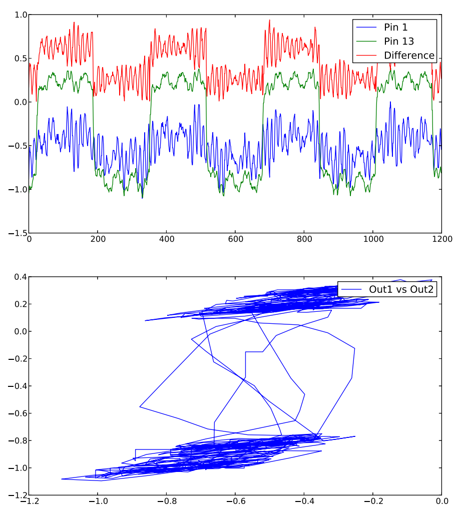

Fig. 23: Material B09S12, 0.53% nanotubes, example with 5 input pins.

total number of states in which a system can be. In case of in-materio computation, the available computational power may be bouded by the number of states that can be read as output from the material. As such, the evolvability range of different material samples plays an important role in the ability to evolve solutions to any kind of computational problem.

B. *System Dynamics*

This subsection presents an observation using the sample with lower SWCNT concentration. Figure 22 shows an example of solution in the randomly generated initial population for the output maximization problem. Here 4 input frequencies are used and the output signals on Pin 11 and Pin 14 are shown in blue and green, respectively. The red line represents the output difference (top image). The bottom part of the Figure shows the XY plot where the two outputs are plotted against each other. Two dense periodic orbits are visible. As such, the resulting output difference may be considered as a non-periodic oscillating pattern. The same result can be observed in Figure 23, where 5 input square waves are used. The results in this section may be considered as an indication that chaotic or semi-chaotic behavior may be achieved within the material, and that such rich behaviors are more likely to emerge in the vicinity of the nanotubes network percolation threshold. Carbon nanotubes randomly dispersed in polymer solutions may be considered as complex networks where a huge number of tiny elements (nano-molecules) interact at a local level and exhibit different emergent dynamics [23]. The idea of complex systems is connected with the notion of "edge of chaos" [11], which may indicate high complexity and computational power. Computation at the molecular level, i.e. computation-in-materio, may be able to produce complex dynamics, as the very essence of the material physics is exploited for computation. New materials that possess such rich properties may be potential candidate substrates for physical implementations of reservoir computing [9], [15].

C. *Physical Properties Exploited by Evolution*

This subsection describes in details some examples of evolved solutions for the maximization and minimization of output difference problem. Figure 24 shows an example solution with 2 input frequencies on the high CNT concentration

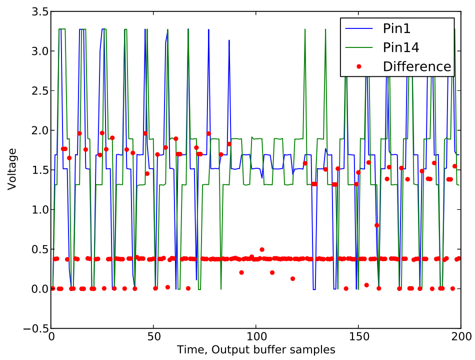

Fig. 24: 5,00% nanotubes, example with 2 inputs, evolve max difference. Input pin 13: 23945 Hz, duty cycle 27%; Input pin 2: 24576 Hz, Duty cycle 96%, Output pins: 1, 4.

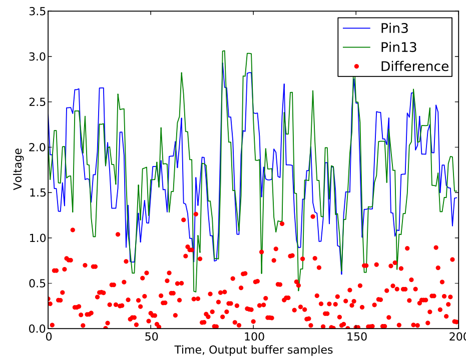

Fig. 25: 5,00% nanotubes, example with 10 inputs, evolve max difference. Input pins (pin#, frequency Hz, duty cycle %): (4, 3205, 49), (10, 23950, 3), (6, 5847, 85), (14, 24761, 46), (15, 8258, 54), (1, 6098, 83), (12, 15177, 40), (2, 19886, 3), (11, 16632, 9), (8, 22859, 99). Output pins: 3, 13.

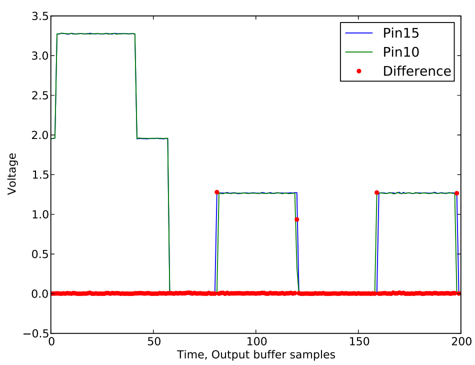

Fig. 26: 5,00% nanotubes, example with 2 inputs, evolve min difference. Input pin 4: 3208 Hz, duty cycle 94%; Input pin 13: 400 Hz, Duty cycle 79%, Output pins: 15, 10.

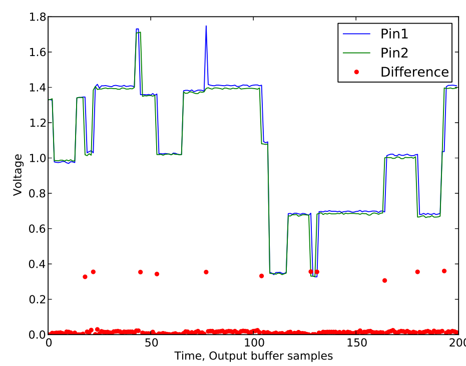

Fig. 27: 5,00% nanotubes, example with 10 inputs, evolve min difference. Input pins (pin#, frequency Hz, duty cycle %): (13, 1976, 78), (9, 3996, 11), (10, 1984, 10), (7, 657, 71), (11, 866, 13), (15, 663, 93), (12, 2060, 41), (6, 24500, 100), (14, 1462, 94), (8, 12910, 100). Output pins: 1, 2.

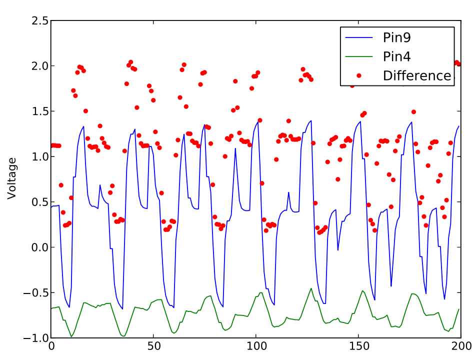

Fig. 28: 0,53% nanotubes, example with 2 inputs, evolve max difference. Input pin 10: 9595 Hz, duty cycle 79%; Input pin 8: 20299 Hz, Duty cycle 91%, Output pins: 9, 4.

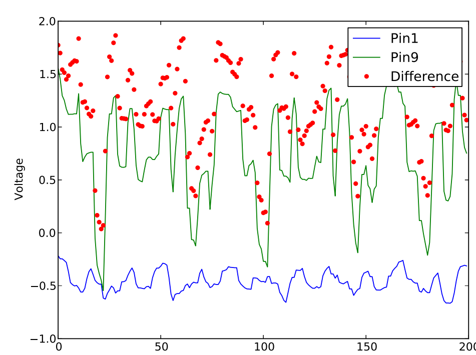

Fig. 29: 0,53% nanotubes, example with 10 inputs, evolve max difference. Input pins (pin#, frequency Hz, duty cycle %): (12, 14190, 13), (8, 22055, 30), (7, 15255, 57), (0, 13302, 100), (11, 15089, 69), (5, 6322, 21), (10, 9437, 72), (14, 7471, 55), (13, 15197, 17), (15, 9928, 29). Output pins: 1, 9.

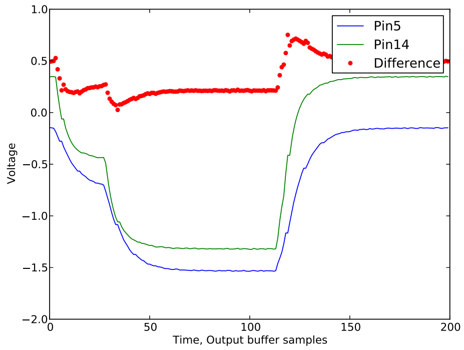

Fig. 30: 0,53% nanotubes, example with 2 inputs, evolve min difference, low frequencies. Input pin 9: 1391 Hz, duty cycle 86%; Input pin 10: 1135 Hz, Duty cycle 17%, Output pins: 5, 14.

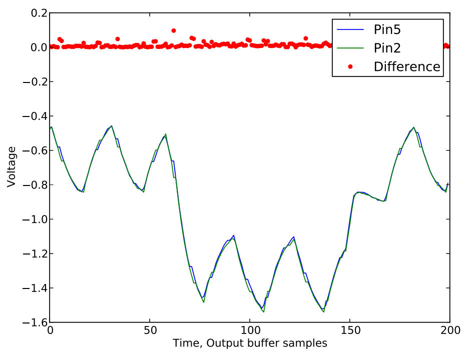

Fig. 31: 0,53% nanotubes, example with 2 inputs, evolve min difference, high frequencies. Input pin 9: 1391 Hz, duty cycle 86%; Input pin 10: 8266 Hz, Duty cycle 17%, Output pins: 5, 2.

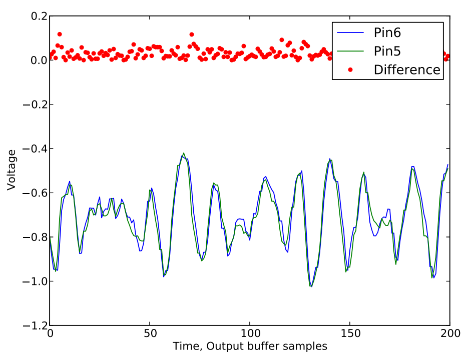

Fig. 32: 0,53% nanotubes, example with 10 inputs, evolve min difference. Input pins (pin#, frequency Hz, duty cycle %): (14, 21177, 100), (15, 23742, 44), (8, 16808, 39), (9, 480, 74), (11, 5246, 88), (10, 17164, 89), (13, 13088, 6), (0, 13358, 89), (4, 10403, 17), (12, 17164, 55). Output pins: 6, 5.

sample for the maximization problem. The green and blue lines represent the two output signals and the red dots represent the difference pattern. It is possible to notice that evolution relies on output signals with similar range of voltage amplitudes, but three interesting phenomena are observed. First, it is possible to notice that one signal is slightly delayed. A plausible explanation for this phenomenon is that the pathways in which the signals travel create a delay line exploited by evolution at the very specific frequencies evolved for this solution. In other words, there are some specific frequencies that enable distinct paths in the material that create delays in the transmitted signals. Note that replacing the material with a straight wire would give the same reading on both output pins, as the set of evolved input signals is fixed and the output values are sampled at the same frequency. A second interesting effect that is clearly visible is a phase inversion. Around sample 80, the green signal has a low peak to 0 V and the blue signal has a high peak to 3.3 V. Finally, a third effect is recognized in the central section of the plot, where there is a signal canceling, i.e. the two signals create a sort of destructive interference. Signal delays, inversions, and canceling are exploited by evolution as other physical characteristics of the materal, i.e. different ranges of electric potential difference, are not available for the chosen sample, as the regularity of the dispersed SWCNT provides more homogenous conductance. The described effects may provide a source of non-linearity in the material. Analyzing Figure 25, it is possible to see that with 10 input frequencies it is much harder for evolution to exploit shifting and inversion. This is reflected in Figure 20, where an increase in the number of input signals makes it harder for evolution to maximize output differences. Looking at the minimization problem on the same sample, both with 2 input signals (Figure 26) and with 10 input signals (Figure 27), the output readings look very similar, e.g. same frequency and same amplitude. Note that the evolved input freqencies are lower than for the maximization problem, in order to avoid possible delays and phase inversions. This is in line with the idea that materials with high SWCNT concentrations have more homogeneous conductive properties.

Different results are obtained with the low concentration CNT sample. In Figure 28, maximum difference was evolved with 2 input signals. The first aspect that is clearly visible is the difference in voltage between -0.6 V and 1.3 V on output pin 9 (blue line) and between -1 V and -0.6 V on output pin 4 (green line). Even if the two output signals are slightly shifted, the major fitness contribution is given by the voltage difference due to the irregular distribution of SWCNT over the electrodes. This means that no matter the used input signals, the voltage output on pin 4 will be always lower than the one on pin 9, due to the physical differences of the material. Without this kind of analysis, one may think that such behavior is a clear strategy discovered by evolution. The same behavior is present when 10 inputs are used, as in Figure 29. In such case, the range of evolvable outputs is broader as evolution can exploit more physical characteristics, e.g. singal amplitudes, delays, and inversions. Not all such physical features are available in the high SWCNT sample. Note that, as shown in Figure 21, adding more input signals does not make the task more evolvable. The input/output relation may be lost if too many input signals are used. Adding more input signals not only increases the search space but also makes the fitness landscape much more complex as more possible pathways are triggered in the material. For the minimization problem, Figure 30 and Figure 31 show two discovered solution using

2 input square waves, the first one with low frequencies and the second one with high frequencies. The solution with high frequencies has higher fitness, as represented by the difference pattern (red dots). Evolution discovered two output pins with similar electrical properties and managed to match the output signals by using high frequencies. This seems unintuitive if compared with the results in Figure 26. The explanation is evident in Figure 30 where lower frequencies were used and the capacitance effect of the material becomes visible. As described in [21], [13], the material holds capacitance. As such, it is possible to notice charge and discharge periods when square wave voltages are used as inputs. Finally, Figure 32 shows the minimization with 10 input samples, where the solution exploits pins with similar properties and fairly high frequencies, if compared with Figure 27.

VI. Conclusion and Future Work

Evolution of physical materials for computation has been used by far as a black box. In this paper we investigated which physical properties are exploited by evolution in order to maximize and minimize differences of output signals. This allowed identifying physical limitations that restrict the range of evolvability in the used materials. As such, we have described the importance of understanding which physical characteristics are available to evolution in order to find out whether solutions are the results of a clear evolutionary strategy or intrinsic constrains due to the material underlying physics. Inspection of evolved solutions showed that the strategies used by evolution to exploit physical properties of material are ofter unanticipated and not intuitive. We observed that materials with lower SWCNT concentrations (yet above the nanotubes network percolation threshold) provide more uneven distribution of nanotubes and polymer molecules, thus allowing a greater range of evolvable output values. This allowed observing rich material dynamics, e.g. towards chaos. Moreover, we identified that evolution in nanotubes materials struggles when too many signals are used, as there is no uniform network within the material and the fitness landscape becomes more complex. As future work we want to use knowledge of exploited physical properties to evolve more stable and repeatable solutions.

Acknowledgment

This research has received funding from the EC 7th Framework Programme (FP/2007-2013) under grant agreement number 317662. Thanks to Kieran Massey and Mike Petty from Durham University for preparing the materials used for our experiments. Thanks to all the partners of the EU project NASCENCE (http://www.nascence.eu) for their collaboration.

References

[1] W. R. Ashby. *An introduction to cybernetics*. Chapman & Hall, London, 1956.

[2] H. Broersma, F. Gomez, J. F. Miller, M. Petty, and G. Tufte. Nascence project: Nanoscale engineering for novel computation using evolution. *International Journal of Unconventional Computing*, 8(4):313–317, 2012.

[3] P. Cariani. To evolve an ear: epistemological implications of gordon pask's electrochemical devices. *System Research*, 10(3):19–33, 1993.

[4] K. D. Clegg, J. F. Miller, K. Massey, and M. Petty. Travelling salesman problem solved 'in materio' by evolved carbon nanotube device. In T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, editors, *Parallel Problem Solving from Nature – PPSN XIII*, volume 8672 of *Lecture Notes in Computer Science*, pages 692–701. Springer International Publishing, 2014.

[5] S. Harding and J. F. Miller. Evolution in materio: investigating the stability of robot controllers evolved in liquid crystal. In *Evolvable Systems: From Biology to Hardware*, pages 155–164. Springer, 2005.

[6] S. L. Harding and J. F. Miller. A tone discriminator in liquid crystal. In *Congress on Evolutionary Computation(CEC2004)*, pages 1800–1807. IEEE, 2004.

[7] S. L. Harding and J. F. Miller. Evolution in materio: Computing with liquid crystal. *Journal of Unconventional Computing*, 3(4):243–257, 2007.

[8] S. L. Harding, J. F. Miller, and E. Rietman. Evolution in materio: Exploiting the physics of materials for computing. *Journal of Unconventional Computing*, 3:155–194, 2008.

[9] H. Jaeger. The echo state approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148:34, 2001.

[10] A. Kotsialos, M. Massey, F. Qaiser, D. Zeze, C. Pearson, and M. Petty. Logic gate and circuit training on randomly dispersed carbon nanotubes. *International journal of unconventional computing.*, 10(5-6):473–497, 2014.

[11] C. G. Langton. Computation at the edge of chaos: phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1):12–37, 1990.

[12] O. R. Lykkebø, S. Harding, G. Tufte, and J. F. Miller. Mecobo: A hardware and software platform for in materio evolution. In O. H. Ibarra, L. Kari, and S. Kopecki, editors, *Unconventional Computation and Natural Computation*, Lecture Notes in Computer Science, pages 267–279. Springer International Publishing, 2014.

[13] O. R. Lykkebø, S. Nichele, and G. Tufte. An investigation of square waves for evolution in carbon nanotubes material. In *13th European Conference on Artificial Life*. Springer, in press, 2015.

[14] O. R. Lykkebø and G. Tufte. Comparison and evaluation of signal representations for a carbon nanotube compuational device. In *Evolvable Systems (ICES), 2014 IEEE International Conference on*, pages 54–60, 2014.

[15] W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

[16] J. F. Miller and K. Downing. Evolution in materio: Looking beyond the silicon box. In *2002 NASA/DOD Conference on Evolvable Hardware*, pages 167–176. IEEE Computer Society Press, 2002.

[17] J. F. Miller, S. Harding, and G. Tufte. Evolution-in-materio: evolving computation in materials. *Evolutionary Intelligence*, 7(1):49–67, 2014.

[18] M. Mohid and J. F. Miller. Evolving robot controllers using carbon nanotubes. In *13th European Conference on Artificial Life*. Springer, in press, 2015.

[19] M. Mohid, J. F. Miller, S. L. Harding, G. Tufte, O. R. Lykkebo, M. K. Massey, and M. C. Petty. Evolution-in-materio: Solving function optimization problems using materials. In *Computational Intelligence (UKCI), 2014 14th UK Workshop on*, pages 1–8, Sept 2014.

[20] M. Mohid, J. F. Miller, S. L. Harding, G. Tufte, O. R. Lykkebø, M. K. Massey, and M. C. Petty. Evolution-in-materio: Solving machine learning classification problems using materials. In *Parallel Problem Solving from Nature–PPSN XIII*, pages 721–730. Springer, 2014.

[21] S. Nichele, D. Laketic, O. R. Lykkebø, and G. Tufte. Is there chaos in blobs of carbon nanotubes used to perform computation? In *7th International Conference on Future Comp. Tech. and Applications, IN PRESS*. XPS Press, 2015.

[22] G. Pask. Physical analogues to growth of a concept. *Mechanisation of Thought Processes*, pages 877–922, 1959.

[23] S. Stepney. The neglected pillar of material computation. *Physica D: Nonlinear Phenomena*, 237(9):1157–1164, 2008.

[24] A. Thompson. An evolved circuit, intrinsic in silicon, entwined with physics. In *1st International Conference on Evolvable Systems (ICES96)*, Lecture Notes in Computer Science, pages 390–405. Springer, 1997.