Computing in Unstructured Matter

Odd Rune S. Lykkebø

21st October 2016

Abstract

This thesis presents an exploratory journey towards new and unconventional material-based computing devices. Conventional computing hardware is the result of a laborious top-down, human-controlled design process where progress has been tracked by Moore's law. However, due to physical and hard limiting factors such as power usage and lack of exploitable parallelism, one can no longer expect this trend to continue. In contrast to this, natural evolution has created information processing architectures ("computers") bottom-up, whose complexity, flexibility, scale and power efficiency outnumber current semiconductor-based technologies. The (unconventional) computing in these systems *emerge* from an evolutionary process. The underlying physical properties of matter in the environment are manipulated or *configured* into physical states that enable computation, possibly through perturbations of self-organising properties of the materials.

The larger context and research problem this thesis can be placed in is to explore these physical properties emerging from a low level, i.e. at nanosystem level. Instead of engineering systems at this level, which is extremely challenging, we take an approach inspired by natural evolution. Using the method of *Evolution-In-Materio*, new computing architectures are built *bottom-up* out of unstructured nanomaterials such as nanoparticles or carbon nanotubes.

We propose a hybrid computer implemented with both conventional and unconventional hardware. An artificial evolutionary algorithm (EA) running on a digital host computer searches for stimulus (*configurations*) that force the material into physical (potentially low level) states in which useful computation can take place. A special custom hardware platform, called "Mecobo", has been constructed in order to generate stimulus into the material and observe its physical response. The platform contains an FPGA used for high-precision sequencing of electric signals with various properties, potentially through analog-digital converters. The host computer communicates with this platform giving the EA direct access to the physical material. In the work presented, the main material used is various *unstructured* samples of randomly dispersed carbon nanotubes in polymers.

The results present an interface between the conventional digital and the unconventional physical world implementing the hybrid computing approach. The Mecobo platform in combination with experiments demonstrate that this method is feasible so as to achieve useful computation. Further, the results show that the physical richness of one unstructured nanosystem instance can be used to solve a variety of task and that methods for state manipulation and physical representations influence a system's ability to produce complexity, allowing for greater evolvability. Finally, some systems have the possibility to induce perturbations to their own dynamics as a function of their system states, enabling state space trajectory changes. Such systems are called. Results indicate that the materials have both a dynamical *electrical* structure and a dynamic response to stimulus, pointing towards the need for a $(DS)^2$ treatment of the materials.

Preface

This doctoral thesis was submitted to the Norwegian University of Science and Technology (NTNU) as part of the requirements for the degree of Doctor of Philosophy (PhD). The Department of Computer and Information Science (IDI) at NTNU has been the host of the work, supervised by Professor Gunnar Tufte.

The research leading to these results has received funding from the [European Community's] Seventh Framework Programme ([FP7/2007-2013] [FP7/2007-2011]) under grant agreement no [317662].

Acknowledgements

My colleagues at the CARD group have been invaluable both as social anchors and research collaboration partners. I wish to deeply thank my advisor, Professor Gunnar Tufte, with his open door and open mind. Without him, I would not have ventured into science. My paper co-authors and thesis correction crew, Stefano, Johannes and Dragana made sure my writing was at least somewhat coherent. I would also like to extend my thanks to the entire NASCENCE consortium, in particular, Simon Harding and Julian Miller whose input and discussions gave me much inspiration and good friendship. For a too short while, I had the pleasure of visiting Professor Takagi's lab at Kyushu University and being one of his students. I still look at the lecture notes.

Finally, my deepest gratitudes goes to my mother Ragnhild, who truly believes, and Mari for being ever patient.

Contents

Chapter 1

Introduction

"I may not have gone where I intended to go, but I think I have ended up where I needed to be."

—Douglas Adams

Today's digital computers represent the culmination of decades of engineering effort. A seemingly insatiable need for ever-faster computation has driven engineering and science further than once thought possible, leading us to incredibly intricate constructions, constantly battling the fine line between forcing too much energy through the system and maintaining stability. The digital computer is a *fragile thing,* and thus stability is important for it to function as expected. So ubiquitous is this digital computer we simply call it *a computer*, though the origins of the term are much older. This is the *conventional* computer. So called because there is a firmly set belief in some that this is what a computer is. *Unconventional computation*, and a variety of other similar labels and scientific fields of research, on the other hand does not deal with how to purely improve the digital paradigm, but rather attempts to explore what a computer *can be.*

1.1 Computation, Evolution, Complexity and Matter

It seems almost negligent not to attempt to understand or replicate the process that has built arguably the most complex systems we know of in the universe, out of the materials which make out the universe: natural evolution. When an individual in nature responds to an environmental event, for example a mouse realizing that an incoming owl from above presents an immediate danger to its survival, the mouse is taking the information obtained by its observational method, i.e. eyes, processing it in possibly complex ways together with other information available to it, forming a calculated or computed response to the information: run away, thus activating its ways of manipulating either itself or the environment by moving its legs. It seems clear that whatever computing ultimately *is*, nature is doing *it*, and the entities involved in the process are products of evolution.

1.1.1 Computation

As Denning notes in his ACM Ubiquity Symposium opening statement [21], the definition of computation has changed as the field of computer science has grown. The relatively old and more conventional models of computation such as the Turing machine, Church's lambda expressions, and Gödel's recursive functions have been joined by more recent nature-inspired models of computation such as P-systems [63], D0L-systems [70], Quantum Annealing [42] and a sleeve of others. Physical systems such as Adleman's DNA computer [5] and Adamatzky's slime mould [1] show us that these unconventional models even have possible physical realizations. Several more examples are mentioned in Chapter 2.

Many conventional models, but certainly not all, are imperative: they describe explicitly the steps required to obtain a certain result. In contrast, many unconventional models and physical systems lean more towards a declarative description: they describe the wanted result but not necessarily the precise steps towards obtaining the result. In the light of the above-mentioned computational models that draw inspiration from nature, it is not a long mental leap to think that nature itself is in some ways doing computation and that the biggest driver for this computation is evolution. In fact, in [76] Toffoli argues that computation simply does not make sense without taking into account the evolutionary feedback loop, and in [75] he argues that nature has some preferred or inherent ways of doing computation. It is impossible to disregard the power of the evolutionary process in nature, though one should note that nature has spent millions of years in an open-ended optimization process, and if we wish to harvest its power we must have a keen understanding of it to accelerate this process.

1.1.2 Evolution and Complexity

Evolution has managed to build highly capable information processing systems in nature. These natural systems can show a fascinatingly rich behaviour, demonstrated for instance by the human brain, rain forest ecosystems, and termite mounds. They are all an integral part of *our* environment. As Strogatz points out when discussing the analysis of linear systems, "*Most of everyday life is non-linear, and the principle of superposition fails spectacularly"* [72]. These systems often show *complex behaviour,* and one of the contemporary fields studying how parts of a system interact with its environment and give rise to global (and potentially complex) behaviour is called *complex systems [66].* The field's roots are old, stretching back at least to the emerging field of cybernetics during the 1950s and 60s, where they were discussed in depth by prominent researchers such as Ross Ashby and Alan Turing in the RATIO club, a dinner club of sorts for academics interested in questions surrounding cybernetics such as adaption and learning, as discussed by Husbands in [37], and Miller et.al. in [53]. Today, *adaptation* is taken to be the "evolutionary process", though this was not immediately obvious. The first cybernetics movement saw adaptation as an important part of *system stability.* Later, scientists such as Arthur and Holland has come to see adaptation as a complexity-building process [7][35], thus linking the study of complexity to the study of evolution and, by the above argument, *computation* . Simulated systems exhibiting complex behaviour such as cellular automata [84] and classifiers, sometimes called Learning Classifier Systems (LCS) [35][34], have been utilized to further study this. Mitchell used artificial evolution to evolve cellular automata for doing 1D density classification [57], and LCS have been used for a number of tasks in a machine learning context, such as medical diagnostics [36]. See [69] for a relatively recent survey. Crutchfield and Langton also studied how certain behaviours of a certain system such as a phase transition from an ordered to a chaotic behaviour do information processing, i.e. computation [40]. The results of these experiments draw up a picture in which evolution is a complexity-building process with a deep link to information processing.

1.1.3 Matter

A physical system is made out of matter. In the NASCENCE project, the geometric organization of the matter is not engineered, but rather the result of more or less random processes, and the relationship between the structural properties and the individual components is not clear. Matter can be described at a number of different scales, from one of the currently known deepest levels as fundamental quantum particles with non-integer *spin*, through atoms and molecules, up to "macroscopic" properties of matter such as electrical conductivity. For each of these levels or models, there is an *aggregation* of information from the level below as seen from a higher observation level, giving rise to observably different behaviour that cannot be derived from observations about local interaction between the parts of the lower-level system. The term *emergence* is an attempt to label this phenomenon *[27].* Langton believed that in order for a substrate or matter to perform emergent computation, it needs to be able to *transmit*, *store* and *modify* information [45].

For most types of matter, Langton's "primitive functions" can be achieved in some form, but it is not always *practical.* At each level of observation there is potential for different paradigms of computation to take place. For instance, computers can exploit quantum phenomena such as spin in carbon at one level, and computers exploiting current-voltage relationships in silicon at another. An interesting question is how evolution affects this relationship. At the topmost level, biological evolution as a process works on complex entities such as animals, seemingly uninterested in the quantum level interaction of fundamental particles; yet this process is still able to produce ever-increasing complexity. It is in this way evolution is often described as a *bottom-up* emergent process, and the work of Evolution-in-Materio is motivated by this.

1.2 Evolution-In-Materio

We started the introduction with a comment about what computers *can be.* One still striking example of this in the context of the previous discussion on matter and computation, is the work of the British Cybernetics duo of Gordon Pask and Stafford Beer. Amongst other experiments, they grew a tone discriminating device or an "ear", by running current through a mix of acidic fluid and silver, causing electrolysis to form silver electrically conducting structures [61]. The material, or substrate, was not embedded with information before the experiments. Pask and Beer controlled the "knobs" that tuned the growth parameters of this system until it seemed to be doing something "interesting" or complex, as we might call it– this example is further discussed in Chapter 2.

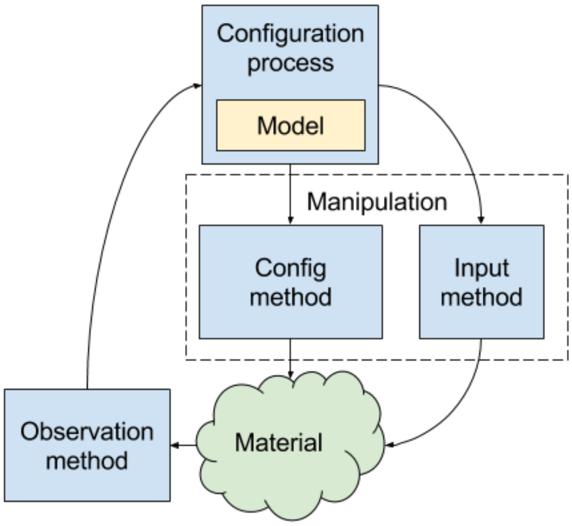


Figure 1.2.1: The abstract unconventional computer in relation to which the research questions are formulated.

Adrian Thompson et.al. took this work one step further. Replacing the human knob-turning with an artificial evolutionary process operating on the configuration data of a configurable hardware chip (FPGA), he evolved such a tone discriminating device by repeatedly trying out the configurations on real, physical hardware and in a live, physical environment [73]. The curious thing about the result, however, was that evolution had surprisingly made a circuit whose mode of operation was not immediately obvious to any human studying the circuit. Evolution had found a novel approach to solve the problem.

The FPGA is a strictly engineered piece of machinery designed to operate within very specific configuration boundaries and its internal geometric structure reflects this. By replacing the FPGA with some piece of material whose microstructures are not human designed we arrive at the method of "Evolution-In-Materio": a *hybrid* approach, in which the observed state for some level of observation of the material is discretised and stored in a conventional digital computer. These observations guide the configuration or "programming" of the material. We call a computing entity evolved in such a manner an EIM hybrid computer, and it is this device this thesis is about.

A physical unconventional computer is a physical entity that exists in some form in nature doing information processing. To obtain computation from this entity one is required to observe its output or state, and this inevitably requires one to choose what information to regard, but also what information to disregard. We shall not concern ourselves with the situation in which the observation method has access to all information. In Shannon-terms, the observation method provides the channel to transmit the processed information in the computer to the outside of the computer. In addition to observation, one must also take the manipulator, or "input" into account. The reason we use the terms manipulate and observe is to not tie our terminology too tightly to concepts which are more in line with the conventional approaches.

An abstract view of an unconventional computer following this structure is shown in Figure 1.2.1. Here we see a configuration process, operating in a feed-back loop together with an observation method and a manipulation method. Note that in EIM, the configuration process is usually some form of artificial evolution. As part of this feedback-loop the configuration process produces a model of the system, which can be implicitly part of an experience-gaining process in which certain types of manipulation have been shown to produce favourable observed results, i.e. as the genetic history of an individual, or explicit as a list of manipulation-observation pairs.

We consider *problem input* separate from the *configuration.* The objective is for the configuration to set the system in some internal state, after which application of input will produce a computational result.

The input and the configuration must both be encoded in some fashion, but we leave open the possibility that the input can either come from some outside source or have a different encoding than the configuration. Using conventional computer terms, the configuration acts as a *program*, but it does *not* necessarily attempt to list steps algorithmically. In a similar sense, the input acts as program *data.* Ultimately, program and data can simply be two different perspectives on the same matter.

1.3 Research questions

The over-arching research question this work has sought to explore and answer is:

> **How viable is an evolution-based hybrid digital-physical unconventional computer architecture?**

Viability can have several meanings. In an engineering context, it means that it is a feasible approach to achieve a specific function, whereas in computer science the viability is connected to obtaining insights and answering questions. In this thesis, we have made gains into answering both of these aspects, and the results and comments in this thesis reflect this.

We have sought answers through the *construction and use* of a hybrid unconventional computer; one where a conventional digital computer and common electronic circuits are used as an interface towards a piece of unstructured material, whose properties are largely unknown apriori. This setup means that since the properties of the material are unknown, the "optimal" manipulation and observation methods are also unknown. We wish to investigate how much can be achieved in computation by using manipulation methods and observation methods that can relatively easily be achieved using traditional electronic methods, i.e. in an integrated silicon circuit (ASIC). To this end, the Mecobo platform has been an integral part of the project and the work documented in this PhD thesis.

Given this contextual question and framework, the following research questions have been explored:

Rq1: What are the effects on the evolutionary configuration process when using different physical intermediate representations in the genotype-phenotype map?

The boundaries between the components of an algorithm in a hybrid system are not obvious. This question seeks to propose where these boundaries can be drawn, and further to explore the mapping between the "pure" abstract form of a configuration in a digital computer, to the "noisy" form emerging when this boundary is crossed. This alludes to development.

Rq2: What are concrete methods utilizing observations of material state to quantify and compare the computational potential in physical computing systems?

While Rq1 focuses on the configuration process involved in the configuration of a *particular* material and the effect of varying the mapping process, we require a method to compare the "potential" or "fitness" for computation across various different materials.

Rq3: What are plausible electrical and computational models for capturing the behavior of a particular material instance capable of solving a particular problem instance?

Given the stochastic nature of the method used to disperse materials, they all have unique geometric structure and hence unique capabilities for solving problems. By investigating models of *electrical* behaviour at various levels we can set lower bounds for the capabilities of one particular material. The additional constraints implied by the formulation of this question are set to establish the importance about the uniqueness of *each* material sample. By investigating *computational* models, we hope to discover features about the *computational* complexity of problems that a material or material instance is rich enough in behavior to solve.

1.4 Thesis outline

This thesis is a collection of papers, which represent the main work and contributions. The remaining parts are organized as follows:

- Chapter 2 gives background and related work.
- Chapter 3 sums up the research process. We present a chronological listing of the articles, and also a brief summary of the engineering work related to the Mecobo digital-physical hybrid computing system.
- Chapter 4 briefly summarizes the articles to be presented in Chapter 6, along with the role of the author. This is to give some background to the contents of Chapter 5 without necessarily having to read the papers in detail.
- Chapter 5 gives some answers to the research questions presented in this introduction. It also includes concluding remarks and future work.

- Chapter 6 contains verbatim copies of all the papers included in this thesis.

In addition, a description of the Mecobo hardware platform has been included in Appendix A. This contains a lot of technical details and may not provide the highest entertainment factor. We hope that it might provide some value for future engineering efforts hoping to build this type of unconventional computers.

Chapter 2

Background and related work

EIM and the Mecobo platform can serve a number of purposes, depending on which questions one seeks answers to. At its core, this thesis centres around an engineered *artefact* and this is primarily what we will focus on in this background chapter as well.

2.1 The NASCENCE project

The NASCENCE (NAnoSCale Engineering for Novel Computation using Evolution) project was an interdisciplinary project whose focus was "... to model, understand and exploit the behaviour of evolving nanosystems (e.g. networks of nanoparticles, carbon nanotubes or films of graphene) with the long term goal to build information processing devices exploiting these architectures without reproducing individual components." [18].

The project was an EU-funded collaboration between York University, The University of Twente, Durham University, Dalle Molle Institute for Artificial Intelligence Research (IDSIA) and The Norwegian University of Science and Technology (NTNU). The work was divided into several "work packages", in which NTNU (and hence this thesis) was to primarily contribute to the construction of the experimental platform hardware and software and to conduct computational experiments with this platform. These packages included the following list:

- *Experimental computation.* Publications include this thesis and number of other publications, such as [19][60][58][43][20][17].
- *Material processing and evaluation*, consisting chiefly of providing thin film slides of materials to the rest of the consortium and gaining more physical understanding of the training process. Publications include [49][50, 52][51].
- *Evaluation and data mining.* Chiefly studied by IDSIA, for instance by Koutnik in[44].
- *Mathematical foundations.* Publications stem for the most part from Twente, and include [16] and [15].

In addition, many reports were produced for the EU commission, freely available at the project web page, *http://www.nascence.eu/* or from the commission directly. An extended list of publications relating to the NASCENCE project can also be found there.

2.2 Cybernetics

Defining cybernetics in contemporary times has become hard, seemingly because of its ability to cover many other fields. In a certain sense it is more of a "way of thinking", a philosophy in which the central dogma is based on the importance of systems displaying feedback-loops. One of the oldest definitions of cybernetics stems from Wiener, who coined it "the science of control and communication, in the animal and the machine" [83]. Wiener's book was one of the first books to attempt to formalize the field of cybernetics, and the book was met with great enthusiasm by researchers in many fields [37]. About the same time in Britain, an informal "dinner club" of sorts named The RATIO club was organized by neurologist John Bates. The club was by invitation only, and is further discussed by Husbands in[37]. Members included scientists and psychologists such as Ross Ashby and Alan Turing, scientists who "*held Wiener's ideas before Wiener's book appeared"*. The meetings occurred monthly, and included topics such as "*Pattern Recognition", "Elementary basis of Information Theory", "The Chemical Origin of Biological Form"* and "*Adaptive Behaviour".*

Richard Ashby went on to write an introductory text book on cybernetics, "An Introduction to Cybernetics" [8][1]. In his book, Ashby wants to shift the focus from studying *reducible* systems whose variable's dependence on each other can be modelled by simple linear relationships, into systems whose behaviour is "so dynamic and interconnected that the alteration of one factor immediately acts as cause to evoke alterations in others, perhaps in a great many others.". Ashby states that this behaviour is *complex*, and he believed *cybernetics* to be the appropriate tool to handle such systems. To demonstrate this interdependence of systems, Ashby did what many cyberneticians did: built a device to exemplify his points. One of the devices, he called the Homeostat.

2.2.1 The Homeostat

The name "Homeostat" derives from "homeostasis", which is the term used for behaviour in which systems display resilience to external perturbations in some environment. Ashby's motivation for building this system was to study how a control system, i.e. the brain and the nervous system could produce adaptive behaviour [9], and provided background to his *law of requisite variety*, which briefly states that the control mechanism must itself have as many possible states as the system it wishes to control. The Homeostat was a simple machine, consisting of four units, each of which contained a magnetic needle representing one of the variables of the system. The position of this needle controlled the amount of current flowing through it. Between the devices, there were variable resistors ("potentiometers", often called simply "pots") and stepping switches (a form of 1-to-many electrical switch) that acted as control parameters. In this way, Ashby could extract general observations about the adaptivity of the system by observing the magnets and manipulating the parameters, which he did by hand.

Ashby saw the Homeostat as a primitive human brain. He believed that the Homeostat's return to equilibrium was a demonstration of the same effect that causes kittens to avoid fire: adaptation and learning. Pickering points out that this is probably one of the first instances of a *real* machine that would "*open-endedly reconfigure itself to respond to its inputs*" [62]. The Homeostat remains relevant though a modern interpretation, continuous time recurrent neural networks (CTRNN) [12].

2.2.2 Gordon Pask and Stafford Beer

Gordon Pask and Stafford Beer were friends, colleagues and early pioneers in cybernetic research, constructing a set of "maverick machines": "*machines that embody theoretical principles or technical inventions which deviate from the mainstream of computer development, but are nevertheless of value"[38]*[62]. The theoretical principles most interesting to Pask and Beer were related to cybernetics and by extension learning and adaptation. Pask's "Musicolour" for instance, was a machine producing various patterns of light in response to sound; and the machine allowed for a certain amount of training or learning. Another example is SAKI, a machine built for automated teaching of operating a punch card machine. These machines were all constructed from conventional hardware, but Pask and Beer were also interested in machines that would learn to interact with their environment, but not necessarily through pre-designed interaction methods. To build these machines, they were searching for materials that were fit for this purpose. One of the criteria they set was that the material had to have enough "variety", as defined by Ashby. They wanted materials that were *inherently* self-organizing, as Beer writes in [13]:

> *If systems of this kind are to be used for amplifying intelligence, or for 'breeding' other systems more highly developed than they are themselves, a fixed circuitry is a liability. Instead, we seek a fabric that is inherently self-organizing, on which to superimpose (as a signal on a carrier wave) the particular cybernetic functions that we seek to model. Or, to take another image, we seek to constrain a high-variety fabric rather than to fabricate one by blueprint.*

Husbands claim that Beer's distaste for electrical circuitry as mentioned in this quote led him to use water fleas, namely Daphna. He would make them ingest small pieces of iron and then attempt to manipulate them using magnetic fields.

It is worth noting that many of the other fields mentioned in this chapter that in some ways relate to Beer's statement above. Some fields are focused on "constructing" this high-variety fabric, whereas some fields are concerned more with utilization through modern methods– nevertheless, the *ideas* themselves are quite old.

A good example of the approach using such "high variety"-materials was an experiment performed by an enigmatic British Cybernetician, Gordon Pask. By inserting platinum electrodes into a iron sulfate solution and running current into the electrodes, iron wires would form along the path of current, however, the acidic fluid would at the same time dissolve these newly formed structures[61]. This dynamic non-stationary behaviour turned out to be possible to manipulate such that it could differentiate between two tones of varying frequency

[1]The title of the book can be misleading. A certain amount of "idea priming" is most likely in order to appreciate its contents.

using a positive reinforcement learning technique. This device can very well be thought of as one of the earliest EIM devices.

2.3 Complex systems

Complex systems is a cross-disciplinary research field encompassing a broad range of topics. The key characteristic of a complex system is how interaction between the components make out the *collective behaviour* of the system through mechanisms such as self-organization and emergence[10]. Another characteristic trait of complex systems includes a focus on *system scale*, i.e. observation levels and the relationship(s) between behaviour at smaller scales and how this affects larger scales. A final often-cited feature is *non-linearity*, where the dependence between variables is not simply a weighted sum.

The study of these systems can be traced back at least to early cybernetics. Ashby explicitly mentions the term "complex systems" in [8], and apart from Cybernetics the field uses a broad range of tools from a diverse range of other fields [66], such as Game Theory[80], Networks [22], Evolution and Adaptation [35], Pattern Formation [78], and Nonlinear Dynamics [72].

By the definition above, many systems can be labelled as complex. The study of how neurons interact, forming how our mind works, is an example of a complex system. The waste-collection system of a city, the internal workings of a cell, the weather, economics and termite mounds: complex systems exist at many scales and in many forms.

There is no agreed-upon common definition of complexity, nor how to measure it. Bar-Yam suggests "*the amount of information required to describe a system"*. Implicitly this requires a set observation level, and for a given observation level or scale (macrostate), the required information is the amount describing the microstate of the system, often called the *entropy.* If the microstate of the system if not directly available, a complexity measure of the macrostate can be done. One such method is *Kolmogorov* complexity, which is a descriptive measure giving the length of the shortest piece of code in a given language giving the output of the system. A way of establishing an upper bound on the Kolmogorov complexity is to compress the output of the system [65].

2.4 Artificial Evolution

Artificial evolution is often interchanged with "evolutionary algorithm", though artificial evolution can be said to be a more general term that encompasses several approaches. Evolutionary algorithms are inspired by (biological) natural evolution. One of the earlier ventures into artificial evolution was the *genetic algorithm,* pioneered by John Holland [33] and further developed extensively by his PhD-student, David Goldberg [26]. The algorithm mimics natural evolution by assigning each individual in a population a score, or "fitness". More fit individuals have a higher chance to breed and produce offspring, carrying more fit genes forward in the generations. Each individual is often represented by a string of either fixed or arbitrary length strings of binary digits, named genes, or *genotype.*

In biology, there is a clearly defined separation between this *genotype* and what is called the *phenotype.* Together with environmental information, the genotype represents the information required for some process to "construct" the phenotype of an individual; its morphology and behaviour. This process if referred to as the genotype-phenotype mapping a term introduced by Alberch in 1991 [6] to enable further discussion on the link between genetics and development. In EA's, this term is used somewhat less strictly in the sense of a function that decodes the genotypic information, such as reinterpreting a string of binary digits into a set of integer numbers (the phenotype). This is called a *direct* mapping, because there is a one to one map between the genotype and the phenotype. By introducing indirect encodings such that the genotype has to pass through some form of development to become an individual can increase the evolvability of a population and the robustness of the individual[81]. The study of these indirect maps or encodings is known as Artificial Development [54][77].

2.5 Unconventional computation

As we mentioned in the introduction, an unconventional computer is essentially any computer that does not fall into the "conventional" camp. As such, any system, device or entity that can be identified as an information processing device but not in a "conventional" fashion belongs under this umbrella. What is conventional or not is a term that change as time passes, but as Stepney notes in [71], there are a number of established "conventional" or classical approaches such as Turing machines and von-Neumann architectures, that in some ways act as distractions towards the goal of *better computing.* In the work of this particular thesis, we are mostly concerned with examples of physical systems used for computation. In this context, we will give a brief *non-exhaustive* listing of some related works here, but note that the field does encompass a very large number of methods, machines, paradigms and theories. What follows is a short list of some of the ones we have drawn most inspiration from in our work.

- **DNA computing** [5], in which Adleman encoded an *instance* of the Hamiltonian path problem in molecules of DNA, and solved it by manipulating it with enzymes and "standard laboratory protocols".

- **Slime mould computing.** In [1] Adamatzky explains the use of the slime mould, a single-cell organism named "Physarum", to solve an instance of the shortest path problem, where the problem is essentially encoded in the pieces of oat that the slime mould uses as nutrients. Adamatzky has also done work on computation using reaction-diffusion systems [3], chemical systems that form complex structures unfolding in time. These systems are also known as Turing systems or "Turing patterns" due to Turing's work on morphogenesis [78].

- **Amorphous computing** [2] saw an opportunity in combining the possibility of creating structures at microlevel (microfabrication) with cellular engineering to create "computing particles" with *local communication* abilities spread in an amorphous fashion, that is, without a clear structure or form.

- **Quantum computing** is a relatively large field and encompasses methods that can be argued to be both conventional and unconventional. The currently most promising are also currently under heavy investigation, for example using quantum-level physical phenomena such as *spin* to do a form of annealing, as commercialized by D-Wave Systems[41].

- **Reservoir computing** is a form of computing where an information reservoir or a low-dimensional input to high-dimensional output map [67], and a *readout*-mechanism (or observation method) such as a quickly-trainable single- or multi-layer neural network, is used to capture the state of the system and use this as output. Various physical systems have been employed as a reservoir, such as a bucket of water [23], where small motors were used to produce waves in the bucket which in turn were used as inputs to a perceptron.

2.5.1 Evolvable hardware

Evolvable hardware is a general term encompassing a few different but related methods, summarized by Greenwood and Tyrell in [28]. As the name implies, they all have some evolutionary method and computer hardware in common, but the purpose of the evolutionary method differs. One method is to use an EA to either design hardware from scratch, for instance by using some level of directly implementable representation of hardware. Such a representation can be a FPGA bitfile or a netlist. The resulting individual from the algorithm is thus an electrical circuit as in [24] and [82]. An alternative method is to implement some form of adaptation into the hardware itself, making it more robust to damaging events or other environmental factors such as in [29].

A second axis on which to place evolvable hardware is to separate between *intrinsically* evolved hardware, in which the hardware has its fitness evaluated *in* real hardware. The advantage here being that all environmental factors are taken into account during evaluation. In contrast, *extrinsic* evolution tests the individuals in a simulated environment, and not in real, physical hardware. One of the biggest advantages here is speed–hundreds of simulations can be run simultaneously, whereas using real hardware can potentially be a time-consuming process, if not down right impossible if combined with the above-mentioned design process and targeting ASIC chips.

2.5.2 Thompson's FPGA experiment

In [73] Thompson describes a thought-provoking experiment. By using a simple EA whose individual's genomes (see section 2.3) were FPGA bitfiles, the file representing the full internal state of the FPGA, Thompson successfully evolved a tone discriminating circuit. When given a frequency above a given threshold, the output of the circuit would go high, and below the threshold, the output would go low. In its own righ, this was an interesting experiment, but it also uncovered potential for exploiting environmental factors in the design of digital electric circuits. Upon examining the result of applying the bit file in a simulator, it became apparent that the circuit would not function in circuit simulation. Further analysis found that even moving the configuration bitfile to other sectors of that same FPGA would not yield a functioning tone discriminator. The EA had found, and exploited, properties of the "black box", that were not "by design".

2.5.3 Liquid Crystal Computing

The main line of thought leading towards the NASCENCE project was done by Miller and Harding. Building on work done by both Mills (see 2.5.6) and Adrian Thompson (see 2.5.2) they used evolution as a tool for evolving a tone discriminator *in-materio* [30]. By building apparatus allowing the application of voltages across terminals in a Liquid Crystal Display (LCD), they were able to find a way to set the state of the material, or *configure it,* such that it operated similarly to Thompson's tone discriminator. Further work included a robot controller in LCD [31] and logic gates in LCDs [32].

2.5.4 Analogue computing

Building devices for performing calculations is not something we discovered this century. Even before the age of Babbage's Difference Engine, the Greeks invented a mechanical *analogue* device for doing celestial calculations known as the Antikythera device [25]. An analogue defines a *systematic relationship* between two concepts [48], such that it is possible to represent one with the other. Another early example of such a relationship can be found is James Thomson's Differential Analyser [74], in which mathematical differentiation was achieved by exploiting the systematic relationship between the relative difference in the rotational speed of two metal discs, forming an integrator, and connecting these together in a feedback loop. The resulting machinery allowed for the calculation of differential equations. With the advent of the operational amplifier, an electric device that neatly encapsulates the concept of a feedback loop, it became possible to use the fact that a capacitor, at a certain level of observation, follows the mathematical equation $V(t) = 1/C \int_{t_0}^{t} I(\tau)d\tau + V(t_0)$, that is, an *integral*, to construct electrical circuits closely following *integration*, which were developed into General Purpose Analogue Computers (GPAC) during the late 1940s to 1950s, and received a mathematical analysis in terms of computational power by Shannon in [68], coining the term GPAC. These integration circuits could be chained together to construct large differential equation solvers. The main drawback of the electrical analogue computer was in practice the variable of integration, time – whereas the analytical solution to the above integral equation does not itself depend on time, an electric analogue computer *must* let the calculation "develop" *using* time, or rather be computed. On the other hand, this was also the strength of the approach. Since the representation of the variables is done by physical entities, the resolution is in a certain sense "infinite". Setting up a systematic relationship between a computation or model and a physical material requires a mapping of both the stimulus of the material and the result of said stimulus, both which can be more or less direct. A direct mapping in this sense is a useful computation that can be obtained by stimulating some material and obtaining a result by observing some property of the output without the use of additional analogues or materials; and an indirect mapping would be the opposite: geometric configurations and composites of materials whose properties gives a given output from a given stimulus. A transistor, when viewed in this light, is a (very small and limited) analogue computer implementing an IF-ELSE-switch-computation by using a particular geometric configuration of silicon and silicon dioxide in combination with electrical stimulus. Today, we use the term "analogue computer" to contrast it with the digital computer.

2.5.5 Rubel's Extended Analogue computer

The GPAC was a theoretical machine. Further analysis of its computational power revealed that it was a limited machine [64]. This prompted Rubel to propose a more powerful, *conceptual*, computer, the Extended Analogue Computer (EAC). Rubel writes in [64], "[...] the extent to which it can be realized by actual physical, chemical, or biological devices or systems remains to be investigated". The EAC essentially consists of hierarchy of levels, each level consisting of a number of "black boxes", such as addition, multipliers, differentiation, and others. The computation happens inside of these black boxes.

2.5.6 Mills's implementation of the EAC

There have been attempts at answering Rubel's question about the realization of the EAC. One of these is J.W. Mills implementation [55]. The board consists of an array of current sources and sinks that provide the input to the material, which is a piece of electrically resistive plastic foam. To measure output, this foam is connected to a set of analogue-to-digital converters. A key property of Mill's EAC is the use of the Łukasiewicz logic array (LLA) [56], which ensures that the EAC can be called a GPAC. The LLA provides an implementation of the basic functions of Łukasiewicz logic, a form of multi-valued logic. Figure 2.5.1 shows one of the 27 functions implemented. V_{min} and V_{max} represent the lowest and highest voltage range observed from the conductive sheet. These voltages are then converted to an "output value" by a microcontroller on the board itself. Mills discretised the functions such that they operated on the range {0, 1/2, 1} scaled to V_{min} and V_{max} .

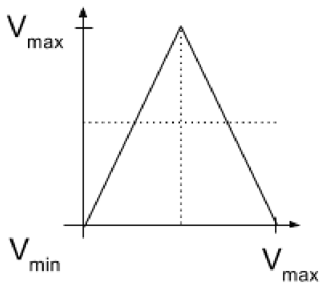


Figure 2.5.1: Łukasiewicz “notch” function, one of the many-valued-logic functions implemented in Mill’s EAC.

To use the foam for computing, the EAC is *configured* by sending a stream of commands through a USB interface. This configuration also sets up if the output is to be passed through an Łukasiewicz logic function, and if so, selects which one. The output of the foam is only read when a “read” command has been sent by the host computer through the USB interface. The value at the desired point is then returned back through this same USB interface by sending a request for data to the EAC.

2.6 Summary

Chapter 2 has presented but a small sample of the many frameworks, models, and systems that one way or another draw inspiration from nature, either in form, process or both. The beauty of using nature as a source is that it is itself always changing, constantly searching for novelty– and nobody needs to tell nature that this is its purpose. A curious side effect of taking inspiration from nature is that it can also increase our knowledge of nature itself; which for many is the one true goal of science as a whole. Chapter 3 gives a birds-eye presentation of the NASCENCE project, which provided the funding and research context for amongst other things this thesis.

Chapter 3

The NASCENCE project

The NASCENCE project was a cross-site, interdisciplinary collaboration project part of the 7th EU Framework Program, "Future and Emerging Technologies" running from 2012 to 2015. Its collaboration partners were The University of Twente, University of York, University of Durham, Dalle Molle Institute for Artificial Intelligence Research and the Norwegian University of Science and Technology. In brief, the scientific contribution of the project was divided into specific work packages distributed "onto" one or several of the partner universities, and they included simulations of EIM systems, mathematical modelling of unstructured nanosystems, experimental approaches to computational properties of EIM systems, data mining and the construction of a platform enabling experimentation with physical computing entities.

3.1 Materials

Several materials have been used in the project. The University of Twente produced a variety of materials based on a mix of CMMA polymers and carbon nanotubes (CNT). The CNT materials were made by two different processes, a drop cast process and a spin coating process. Later in the project Durham also provided liquid crystal-mixes with CNTs. Most of the CNT experimental work was done at room temperature. The University of Twente used gold particles in a cryogenic chamber to lower the temperature allowing phenomena like the coulomb blockade to occur.

Volpati and Massey explored the use of magnetic and electric fields to move CNTs dispersed in liquid crystal [79, 52].

3.2 Mecobo experimental system

The contributions in this thesis fit into the "computational experiments" and "experimental platform" package of the project work packages mentioned above. The latter of these contributions proved to be a time- and

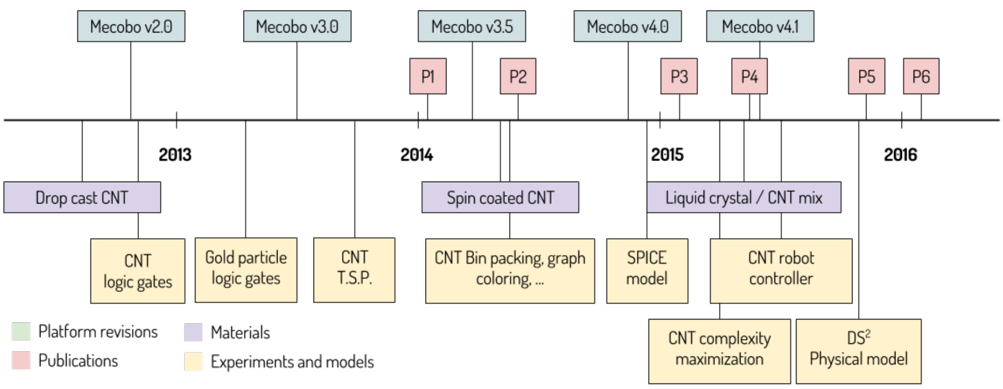


Figure 3.0.1: Project timeline

effort-consuming part of the project as a whole.

Lykkebø, Tufte and Miller had as part of Lykkebø's Master's Thesis [47] designed an early prototype of what would become this experimental system, which we refer to as "Mecobo" in this thesis. This prototype was part of the original project proposal.

A picture showcasing all the produced boards can be seen in Figure 3.2.1. For each board we produced around 5-8 working copies, and unfortunately broke several as well. The process of building these experiment systems was a "design by meetings". Early in the project we held several meetings where all partners where represented, in which the experimental system design was done.

The drive for using electrical voltages as the manipulation method was first of all motivated by the *hybrid* approach of EIM, as mentioned in the introduction. It was further motivated by early experimental work done by Tufte on CNTs and the work done on Evolution-In-Materio by Harding and Miller in [31] and [30], and the need for a method that would allow working remotely with the system rose as a natural requirement with respect to the geographic distribution of the partner universities. Another natural defining boundary we chose to work within, was a focus on *unstructured* (randomly dispersed) samples of Single Walled Carbon Nanotubes, SWCNTs, which served as a useful limiter of scope; making the focal point of the research smaller.

The work continued iteratively with periodic meetings defining synchronization points in which the partners would come together and discuss results and possible future directions to take. As a natural part of this incremental work flow, the Mecobo platform would mutate and adapt to the needs of the group.

Most of the early work, in which we also took part, focused on repeating previous experiments involving boolean logic, from the lowest level in the NASCENCE project to the higher levels in terms of observation level and computational experiments and abstraction.

An example of the lowest level we worked within, nanoscale, is [17], in which Bose et al. used "single electron transistors" to implement boolean logic gates. On the other end of the scale, Massey and Kotsialos found ways of training the material using static voltages to produce stable logic gates [50, 43], and we presented our platform and used it to produce a XOR gate in **P1.**

Concurrently, work on electrical properties of the CNTs themselves was also done by Massey in [51, 49], which informed the discussions for the future experimental platforms.

As the Mecobo platform was produced in Trondheim and distributed to the partners, experimental work began with attempting to expand the computational tasks beyond those of individual logic gates, questions about representation and the encoding of information in the manipulation method came up.

The Mecobo system was built to be expandable, and the need for an "analogue expansion board" that could produce and measure analogue voltages (as opposed to only digital voltages) was designed and produced by NTNU, leading up to an investigation of representations done in **P2**. The use of square waves was carried further in a computational context in work by Mohid et. al. where we sought to move away from the boolean logic-approach and attempt to solve more complex computational problems such as bin-packing and function optimization [59, 60]. The square waves were further investigated systematically in **P3**, where we also identified a SPICE model representing a *static* view of the material. This model was used to replicated the results of Clegg and Miller in [20] in which they solved an instance of the travelling salesman problem in-materio. Finding this observation level of the material somewhat limiting, we decided that the next iteration of our part in the project would revolve around dynamic properties of the materials.

The link between complex output and computational power further motivated lifting the abstraction level from a per-problem view to a view in which we consider generating complex output a goal in and by itself, as discussed in **P4** and in Chapter 1.

Further work on the Mecobo platform was then required, as we came to realize that decisions around implementation done in early stages such as the use of millisecond scheduling resolution would no longer suffice in a context investigating the dynamical properties of the materials. A larger demand on scheduling stability was also required to increase our confidence in the reproducibility of our results. This prompted a re-design of the previous digital logic and buffering systems, resulting in Mecobo 4.1, as it currently stands at the end of the project.

This version allowed us to produce the experimental results of **P5** and **P6**, whose focus was on which *dynamic* properties of the materials were observable and exploitable for evolution.

3.3 Collaborative research

The NASCENCE project had a very clear mission statement, which provided useful directional information in terms of research questions. The further division into the above-mentioned work packages gave each individual researcher associated with a package further direction, while simultaneously limiting the scope of the research to be performed by this individual which may or may not always align with the individual's primary research interests.

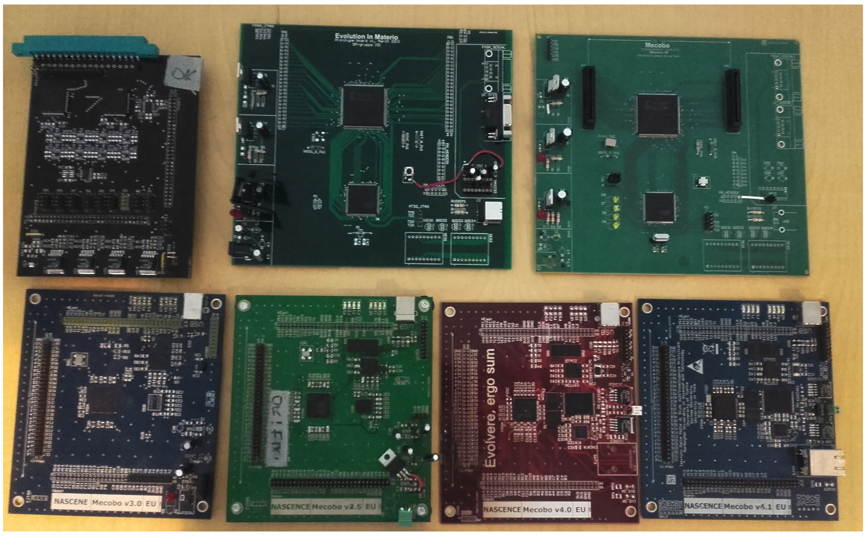


Figure 3.2.1: The Mecobo Experimental platforms and the analogue daughterboard, built by Lykkebø and Tufte during the NASCENCE project. Top row, right to left: daughterboard, version 2, version 1. Bottom row, right to left, version 3.0, version 2.5, version 4.0 and the final version before the project ended, version 4.1.

Finally, as with most things, the scientific process requires ample amounts of time to consume and distil the core ideas presented in research, and in the grand scheme of things, three years is not a very long time to obtain great progress and in that sense we believe the results, and not to mention the experience obtained from the NASCENCE project can certainly be used as a starting point for further research.

Chapter 4

Paper and research result summaries

4.1 P1: Mecobo: A Hardware and Software Platform for In-Materio Evolution

Odd Rune Lykkebø, Simon Harding, Gunnar Tufte, and Julian F. Miller
Unconventional Computation and Natural Computation: 13th International Conference
Springer 2014

Role of authors

Lykkebø wrote most of the paper, coded and ran the experiments. The additional authors were part of the design process and the initial idea of the built system (the Mecobo platform), in particular, Simon Harding designed the Thrift interface for NASCENCE, through which all network and local communication with the Mecobo platform was done.

Abstract

Evolution-In-Materio (EIM) exploits properties of physical systems for computation. "Designs" are evolved instead of a traditional top down design approach. Computation is a product of the state(s) of the material and input data. Evolution manipulates physical processes by stimulating materials assessed in situ. A hardware-software platform designed for EIM experimentation is presented. The platform, with features designed especially for EIM, is described together with demonstration experiments using carbon nanotubes in a thick film placed on micro-electrode arrays.

Retrospective

Our goal for the system was an interface to a black box, allowing a large amount of manipulation methods to be tested. However, since the potential inputs to the material are unbounded, it was impossible to build the perfect apparatus that would cover all types of materials we would encounter during the project.

To account for this intractability we made the system extensible via add-on boards, such as the "ADA daughterboard" that extended the capabilities of the motherboard with analogue-to-digital and digital-to-analogue converters. The paper, however, uses only the basic capabilities of the motherboard; which is to manipulate the material through square waves with varying duty cycles and observe through *digital sampling,* essentially connecting a subset of the conductors into the material to flip-flops in the FPGA and reading out their state periodically.

The paper demonstrates that the Mecobo platform is capable of producing non-trivial computation by using an evolutionary algorithm to find a XOR gate.

4.2 P2: Comparison and Evaluation of Signal Representations for a Carbon Nanotube Computational Device

Odd Rune Lykkebø and Gunnar Tufte
IEEE International Conference on Evolvable Systems 2014
IEEE 2014

Role of authors

Lykkebø wrote the main parts of the paper, had the main idea and ran the experiments. Tufte contributed through discussions, and paper corrections.

Abstract

Abstract—Evolution-In-Materio (EIM) exploits properties of physical systems for computation. Evolution manipulates physical processes by stimulating materials by applying some sort of configuration signal. For materials such as liquid crystal and carbon nanotubes, the properties of configuration data is rather open. In this work, we investigate what kind of configuration data that most likely will be favourable for a carbon nanotube based device. An experimental approach targeting graph colouring is used to test three different types of signal representation: static voltages, square waves and a mixed signal representation. The results show that all signal representation was capable of producing a working device. In the experiments, square wave representation produced the best result.

Retrospective

This paper was written soon after P1 and was a first investigation into the effects of the observable properties of the material when manipulated by three different ways of encoding a configuration of the material. A secondary objective was to investigate if there were obvious differences in the so-called “evolvability” of the configuration representation. The experiment was done by attempting to solve a very simplistic version of the graph-colouring problem using a simple graph and 3 colours.

The problem in and by itself was not meant as an “application”, but rather a vehicle for showing that there were differences that need to be taken into account. This was also the first paper the notion of observation level was made clear, and we made a first attempt at defining the internal boundaries of the system.

4.3 P3: An Investigation of Square Waves for Evolution in Carbon Nanotubes Material

Odd Rune Lykkebø, Stefano Nichele and Gunnar Tufte
European Conference on Artificial Life 2015
MIT Press 2014

Role of authors

Lykkebø had the main paper ideas and ran the experiments on the Mecobo system and the SPICE model. Nichele contributed with additional experiments, discussions, and paper corrections. Tufte contributed to the SPICE model ideas and discussions.

Abstract

Materials suitable to perform computation make use of evolved configuration signals which specify how the material samples are to operate. The choice of which input and configuration parameters to manipulate obviously impacts the potential of the computational device that emerges. As such, a key challenge is to understand which parameters are better suited to exploit the underlying physical properties of the chosen material. In this paper we focus on the usage of square voltage waves as such manipulation parameters for carbon nanotubes/polymer nanocomposites. The choice of input parameters influences the reachable search space, which may be critical for any kind of evolved computational task. We provide common measurements such as power spectrum and phase plots, taken with the Mecobo platform, a custom- built board for Evolution-In-Materio. In addition, an initial investigation is carried out, which links the frequency of square waves to comparability of the output from the material, while also showing differences in the material's physical parameters. Observing the behaviour

of materials under varying in- puts allows macroscopic modelling of pin-to-pin characteristics with simple RC circuits. Finally, SPICE is used to provide a rudimentary simulation of the observed properties of the material. This simulation models the per-pin behaviours, and also shows that an instance of the travelling-salesman problem can be solved with a simple randomly generated cloud of resistors.

Retrospective

The work in **P2** prompted us to look more closely into the observable physical properties of the material since it became clear that if were going to use square waves as a configuration method, the search space expanded dramatically and we would like to have further information about this method's particular impact on the CNTs.

An investigation of square waves was performed, finding that the material response resembled that of a low pass filter. This further spawned the idea that we could potentially use SPICE as a model of the system. The take-away from the paper was that by using a model of the material, i.e. a SPICE circuit consisting of randomly connected low-pass filters, we could solve a certain problem, the circular travelling salesman problem. Thus the material is *at least* as rich in behaviour as a randomly connected resistor network, interspersed with capacitors.

4.4 P4: An Investigation of Underlying Physical Properties Exploited by Evolution in Nanotubes Materials

Stefano Nichele, Odd Rune Lykkebø, and Gunnar Tufte

IEEE Symposium Series on Computational Intelligence
IEEE 2015

Role of authors

Nichele did the experimental work and had the main paper idea. Lykkebø contributed to the idea formation phase, experimental work and discussions along with minor paper contributions. Tufte contributed with technical know-how, discussions, and paper corrections.

Abstract

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 underlying 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.

Retrospective

The evolvability of the EIM systems remained a key focus as we further investigated the challenges of having a “black box” in the evolutionary feed-back loop. P3 used human intuition to develop a model of the material. This paper, on the other hand, attempted to use evolution of a relatively simple task as a way to establish some more “physical limits”. The findings showed a connection between lower concentrations of CNTs allowed a greater range of observable dynamics; and further that the amount of energy pushed into the material by the manipulation method, i.e. the number of input signals, was a “hard” limiting factor on the evolvability.

4.5 P5: Evolution-In-Materio of a dynamical system with dynamical structures

Odd Rune Lykkebø, Gunnar Tufte

International Conference on the Synthesis and Simulation of Living Systems 2016
MIT Press 2016

Role of authors

Lykkebø had the main paper idea, did the experiments and wrote the main part of the paper. Tufte contributed with discussions, the $(DS)^2$-contextualization idea and paper corrections.

Abstract

Evolution-In-Materio aims to exploit real-world physics of materials to achieve computation by a combination of external stimulus and interpretation of the state of materials through measurements and observations. In a majority of Evolution-In-Materio work the dynamics of the material is filtered out, or the problem is defined in a way that the sought solution is a point attractor. In this work we explore the dynamics of materials. Within the assumption that suited materials include rich behaviour emerging from the under- lying physical processes there should be observable behaviour similar to Dynamical Systems with Dynamical Structures ($(DS)^2$). Such behaviour results in systems with a possibility of inducing perturbations to their own dynamics. Further, the importance of the observation level used when observing and interpreting the state of the materials is discussed and related to dynamics in Evolution-In-Materio systems

Retrospective

The dynamical nature of the materials along with a limited resolution in time (i.e. sample rate) motivated us to look into increasing the spatial resolution of the system; i.e. allowing more measurement probes to be connected. Due to practical reasons and limits of the Mecobo platform, this excluded our ability to do analogue-to-digital conversion of the signals and we were left with a simple mechanism in which we used the trigger voltage of flip flops to do sampling. This also had the added benefit of higher sample rates. That the material shows a$(DS)^2$-like behaviour means that the "search space" in which we look for configuration data broadens tremendously and the methods and tools we employ *must* take this double-dynamic nature into account. At the same time, this points to even further richness of at least two materials (common salt and CNTs) and an even larger potential for obtaining useful computation from the system.

4.6 P6: Dynamics in Carbon Nanotubes for In-Materio Computation

Stefano Nichele, Johannes Høydahl Jensen, Dragana Laketić, Odd Rune Lykkebø and Gunnar Tufte
International Journal On Advances in Systems and Measurements, Volume 9
IARIA 2016

Role of authors

Nichele and Jensen did the main experimental work on this paper. Laketić provided text and work on the theoretical work. Lykkebø contributed with ideas relating to chaotic systems and structured analysis of the materials, discussions and paper corrections. Tufte contributed with ideas, discussions and paper corrections.

Abstract

In-materio computation exploits physical properties of materials as substrates for computation. Evolution-In-Materio (EIM) uses evolutionary search algorithms to find such configurations of the material for which material physics yields desired computation. New unconventional materials have been recently investigated as potential computational mediums. Such materials may intrinsically possess rich physical properties, which may allow a wide variety of dynamics. However, how to access such properties and exploit them to carry out a wanted computation is still an open question. This article explores the dynamics in one particular type of nanomaterials which is used to solve computational tasks. Nanocomposites of Single-Walled Carbon Nanotubes (SWCNTs) and Poly-Butyl-Meth-Acrylate (PBMA) are configured so as to undergo evolutionary processes with the goal of performing certain computational tasks. Early experiments showed that rich dynamics may be achieved, which may yield complex computations. Some indications of chaotic behaviour were observed so further work was carried out with the aim of examining the dynamics achievable by such nanocomposites. Since it is not an easy task to access the physics at the very bottom of the material, investigation of the material dynamics is kept within the limits imposed by our measurement equipment and the level of observability enabled by it. Presented results show that interesting, complex dynamics is achievable by examined nanocomposites and that it depends on the type of signals used for the material configuration as well as on the material intrinsic properties such as percentage of SWCNTs in the nanocomposite.

Retrospective

This paper stands closely together with P4 as a "systematic inquiry" into the observed properties of the material; which were shown to be inherently *dynamic* in nature. The results in the paper provide a strong indication that the material properties add "value" or complexity to the input data. Further results also confirm the findings in P4 where a limit is found with respect to the amount of inputs to the system.

4.7 Other papers

Other papers in which I have had some contribution is listed below:

- Nichele, Stefano; Laketić, Dragana; Lykkebø, Odd Rune Strømmen; Tufte, Gunnar. (2015) Is There Chaos in Blobs of Carbon Nanotubes Used to Perform Computation?. FUTURE COMPUTING 2015, The Seventh International Conference on Future Computational Technologies and Applications.
- Laketić, Dragana; Tufte, Gunnar; Lykkebø, Odd Rune Strømmen; Nichele, Stefano. (2016) An Explanation of Computation - Collective Electrodynamics in Blobs of Carbon Nanotubes. 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS).
- Laketić, Dragana; Tufte, Gunnar; Nichele, Stefano; Lykkebø, Odd Rune Strømmen. (2015) Bringing Colours to the Black Box - A Novel Approach to Explaining Materials for Evolution-In-Materio. FUTURE COMPUTING 2015, The Seventh International Conference on Future Computational Technologies and Applications.
- Mohid, Maktuba; Miller, Julian; Harding, Simon; Tufte, Gunnar; Lykkebø, Odd Rune Strømmen; Massey, Mark K.; Petty, Michael C.. (2014) Evolution-in-materio: Solving function optimization problems using materials. 14th UK Workshop on Computational Intelligence (UKCI).
- Mohid, Maktuba; Miller, Julian F.; Harding, Simon; Tufte, Gunnar; Lykkebø, Odd Rune Strømmen; Massey, Mark K.; Petty, Michael C.. (2014) Evolution-In-Materio: Solving Machine Learning Classification Problems Using Materials. Lecture Notes in Computer Science. vol. 8672.

Chapter 5

Results, conclusions, and future work

"One describes a tale best by telling the tale."

— Neil Gaiman, *Fragile Things*

In this final chapter, we will cover results obtained from the papers included in the thesis and how they relate to the research questions. Note that in a "blue skies" project like NASCENCE ultimately is, conclusions should be made with care and read in the context of the papers in which they are made.

5.1 Rq1: What are the effects on the evolutionary configuration process when using different physical intermediate representations in the genotype-phenotype map?

The common understanding of "phenotype" in Evolutionary Algorithms is as the result of the genotype-phenotype map, in which some interpretation of the genome data is done on the genome before it is evaluated, *with full information,* in the fitness function.

In EIM, we have crossed the "reality gap"[39] by beeing fully intrinsic, and are therefore left with partial information, i.e. observations of the behaviour of some *entity.*

Further, since the observations and manipulations are both done electrically, the observations can also potentially affect this entity itself.

To aid discussion and further work, we have found it useful to *define* the phenotype as a time-varying entity, existing from the moment access is made to the digital data structure holding the genome information, until the end of the application of configuration and problem input. The observations inevitably become part of the phenotype, as well as steering the configuration process by providing the partial information available to the fitness function.

- The genotype-phenotype map consists of (at least) two parts: the first takes the digital representation and translates it into some abstract representation of a physical signal, i.e. "square wave at 20KHz", and the second part takes this abstract representation and produces the actual physical signal using real physical hardware, i.e. the ADC converter, the traces leading into the material and the power supply units.

- As **P2** shows, there are differences in the obtained configurations when the EA operates directly on voltages and frequencies. The result is a small but noticeable difference in the obtained fitness when operating on square waves. Unfortunately, the genome did not use the same "digital representation" in this experiment, making direct comparisons between the physical representation harder.

- By defining the phenotype as observations during a time-period, the timespan of the phenotype becomes a parameter that scales the "search landscape" in the evolutionary process. The experiments in P4 shows that by using square waves in combination, the search landscape becomes less predictable and more complex. In this sense, the configuration process slows down.

5.2 Rq2: What are concrete methods utilizing observations of material state to quantify and compare the computational potential in physical computing systems?

In the NASCENCE work we have used two approaches to this question.

- *Systematic investigation by human intuition*: The gate-sweep of **P1**, detailed analysis of waveforms in **P3** and frequency sweeps of **P6** reveal manipulation parameters such as geometric region, or rather the pins or probes connected to a particular region, and signal parameters where the material is more reactive to manipulation.

- *Searching*: By using search strategies seeking to maximise some measure of complexity. As shown in **P4** and **P6**, using this approach revealed that the manipulation of the material interact *globally* to generate more complex state observations. We have also found time-dependent transition behaviours as we demonstrated in **P5**.

5.3 Rq3: What are plausible electrical and computational models for capturing the behaviour of a particular material instance capable of solving a particular problem instance?

- SPICE modelling can be used to produce an electrical model capable of solving the same problem instance as a material, as we show in **P3**. Note that we did not attempt to create a model that matched one specific material; rather we show that our model is *at least* as rich or *varied* as the material, for this particular observational level.

- As for computational models, as **P5** shows, the EIM system can be described as a physical realization of a $(DS)^2$ system. This implies that a *computational* model of this system should take the dynamical structure of the system into account. Though we did not pursue the use of the MGS language for writing computational models in this paper, we believe this is a logical next step.

5.4 Conclusions and future work

Finally, we return to our initial research question.

> **How viable is an evolution-based hybrid digital-physical unconventional computer architecture?**

The articles in this paper have attempted to explore the possibilities of material-based physical unconventional computers created by the EIM hybrid approach. The viability of the proposed computing paradigm depends on the definition of viable. As a direct replacement for the digital conventional computer, the EIM approach certainly can not be said to be viable based on the results of this thesis. However, as a way to explore unconventional computing, we have come some way, and no matter how modest our results, we believe the approach carries value not found in the context of conventional computers.

5.4.1 Crossing the hybrid boundary

The viability of any computer architecture depends on finding a *trade-off point* between the amount of energy that is used for manipulation, the energy used for observation and the energy used for the virtual-to-physical (and potentially the reverse) mapping of information.

Assuming that the result of a computation is to be somehow useful for further work; some form of decoding of information must take place. One could imagine a material with some possibility of external morphological change to take the form of the answer of the computation, i.e. as in the case of Adamatzky's slime mould where the produced mould is e.g. a map of a road network [4], ready for consumption by a human brain– however if this is to be part of some feedback optimization loop, the boundary into the virtual world of the digital computer must be crossed by some medium such as a camera and image (information) processing decoding the real physical result to a digital result. An even "purer" approach entails direct observations of material state which can be fed into another physical computer. However, at some point, *in a hybrid architecture*, the boundary must be crossed.

Within the constraints placed by the hybrid model, the approach chosen by us in P5 seems to be the "path of least resistance" in the sense that evolution is forced to adapt to the observation method which can be used directly in a digital computer, the triggering of flip flops (registers) which in a certain sense is the very core of a digital system. This approach becomes very attractive when one considers integration into an ASIC chip.

In the case of the EIM unconventional computer, the macroscopic electrical properties of the material impacts the complexity and potentially the computational power of a particular material. CNTs have relatively high DC resistive properties, and the amount of CNTs in the material sample is a macroscopic property on the overall reactance of the material. However, since we do not control the geometric structures, a high amount of CNTs implies higher conductivity on the pins of the material; effectively creating all-to-all electric conducting connections between pins. The net result of this is that since we are using a low number of pins, the amount of "interesting" localized behaviour is limited. Due to the voltage range being relatively fixed in our hybrid system, we must control the resistive property to allow current to flow in the material.

Finally, processing information implies the *engagement* of the material, and in a real physical system there is a great number of potential sources of "false positives", any part of the whole system can contribute to the processing, even the environment.

5.4.2 Time

When interfacing a physical system, time can no longer undergo "compression" as it does when simulated in a purely virtual realm as is common in artificial evolution. However, for a computation to happen "in-materio", or "in-real-life" as it were, one of the required resources is *time*. The computation must be allowed time to "unfold" and this effect is amplified if structural reorganization of the material happens.

One of the points not explored explicitly by our research is how long this time period need to be, however, it seems clear that it is highly dependent on both the abstract nature of the computation, i.e. the problem, the material used and the apparatus used for the manipulation and observation. As noted by P4 and P5, the state of the material changes over time. The dependence on the problem is due to having to produce the actual manipulations required for that particular problem, the material dependence arises from the physical response to the manipulation taking different amounts of time.

Often in artificial evolution, we see thousands of entities being evaluated in each generation. In the virtual realm, this is trivial to do in parallel, dividing the time required by the number of computing units available. However, this is impossible in the physical realm, as mixing stimulus would interfere with the observations of the state of the system. A way around this is to use a model of the system, such as the one proposed in P5, however, this defeats the purpose of exploiting *unknown* properties emerging from manipulation-material-interactions. Another possible solution is to build up a library of "primitive" behaviours and attempt to map the information (both problem and input) to the physical manipulation producing these behaviours or to build up a virtual layer on top of the physical system. At any rate, it is clear that the configuration must happen "off line" to be of use, at least in a conventional context.

Perhaps there are ways of rethinking the framework such that comparison with conventional computing architectures is invalid.

5.4.3 Future: What can an unconventional computer look like?

There is essentially two main directions to pursue. The first is within engineering and material science:

- Following Pask and Beer in the search for materials of "sufficient variety". One key feature of these materials can be the capability of having some observable *state* (i.e. memory), and a deterministic shape and behaviour in response to an external stimuli. One obvious candidate is *neurons*. Neurons show tremendous potential for growth and adaptivity, as demonstrated by the human brain.

- Continuing the development and analysis of the *interface* towards the materials, and to further investigate the impact of stimulus- and observation methods on the overall computational capabilities of the system. In the context of neurons, this must also include an interface that keeps the cells alive, and provides ample growth opportunities.

Secondly, it is possible to see a possibility of *programming* the systems manually, as opposed to using an EA explicitly, and working with computational models of material computers.

- In Quantum Annealing, a single problem has been chosen as "the one" in which computation of many forms is cast into [42]. The same could be done with Evolution-In-Materio systems; for instance by formulating the problem as an optimization problem in which evolution is used to find the set of minimal measurements satisfying some predetermined requirements.

- In contrast, a more general approach involves a programming language and some method of implementing the semantics of the language. Literature is rich with proposed languages and models of computation that draw inspiration from physical processes, such as L-Systems [46], P-Systems [14] and Membrane Computing [63]. Creating a computational model inspired by for instance manipulating materials with square waves, and mapping the behaviours of the material to "primitive functions" which can be used to build higher-level functions. Genetic programming could further be used as a method to program these systems. In this way we lift our view from an entirely per-application view of the material to a more general "virtual machine" type of view, one where the "Instruction Set Architecture" of the computer is implemented in some material.

An obvious candidate material for further research is a collection of *neurons [11]*. Neurons are electrically excitable, and are thus potentially a good fit in a digital-physical hybrid computer, as long as we can manage to maintain a live population of them. Neurons further interact locally with neighbouring cells through growing and modifying connections, called *synapses*, to adapt to new stimulus, for example acquiring new memories or learning new tasks. This behaviour fits well within a $(DS)^2$ systems view, since there is both dynamics *of* and *on* networks. Our initial work on these systems forms a basis for further investigation. Approaching this from a computer science perspective could even add to understanding how synapses form in response to computational tasks and help to further investigate the link between physical systems and computation.

Bibliography

[1] Adamatzky A. *Advances in Physarum Machines*. 2016.

[2] Harold Abelson, Don Allen, Daniel Coore, Chris Hanson, George Homsy, Thomas F Knight Jr, Radhika Nagpal, Erik Rauch, Gerald Jay Sussman, and Ron Weiss. Amorphous computing. *Communications of the ACM*, 43(5):74–82, 2000.

[3] Andrew Adamatzky, Benjamin De Lacy Costello, and Tetsuya Asai. *Reaction-diffusion computers*. Elsevier, 2005.

[4] Andrew Adamatzky and Jeff Jones. Road planning with slime mould: if Physarum built motorways it would route M6/M74 through Newcastle. *International Journal of Bifurcation and Chaos*, 20(10):3065–3084, 2010.

[5] Leonard M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(11):1021–1024, nov 1994.

[6] P Alberch. From genes to phenotype: dynamical systems and evolvability. *Genetica*, 84(1):5–11, 1991.

[7] W. Brian Arthur. On the Evolution of Complexity. Technical report, nov 1993.

[8] W Ross Ashby. An Introduction To Cybernetics. *Director*, 80(4):295, 1999.

[9] Wl Ashby. *Design for a brain: The origin of adaptive behaviour*. Springer Science & Business Media, 2013.

[10] Yaneer Bar-Yam. *Dynamics of Complex Systems*, volume 2. 1997.

[11] Mark F Bear, Barry W Connors, and Michael A Paradiso. *Neuroscience*, volume 2. Lippincott Williams & Wilkins, 2007.

[12] Randall D Beer. Parameter space structure of continuous-time recurrent neural networks. *Neural Computation*, 18(12):3009–3051, 2006.

[13] Stafford Beer. A progress note on research into a cybernetic analogue of fabric, 1994.

[14] Gerard Berry and Gerard Boudol. The Chemical Abstract Machine. In *Proceedings of the 17th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, POPL '90, pages 81–94, New York, NY, USA, 1990. ACM.

[15] Paul Bonsma. Independent Set Reconfiguration in Cographs. In Dieter Kratsch and Ioan Todinca, editors, *Graph-Theoretic Concepts in Computer Science: 40th International Workshop, WG 2014, Nouan-le-Fuzelier, France, June 25-27, 2014. Revised Selected Papers*, pages 105–116. Springer International Publishing, Cham, 2014.

[16] Paul Bonsma, Marcin Kamiński, and Marcin Wrochna. Reconfiguring independent sets in claw-free graphs. In *Algorithm Theory–SWAT 2014*, pages 86–97. Springer, 2014.

[17] S K Boses, C P Lawrence, Z Liu, S Makarenko, R M J van Damme, H J Broersma, and W G van der Wiel. Evolution of a designless nanoparticle network into reconfigurable Boolean logic. *Nature Nanotechnology*, 10(12):1048–1052, 2015.

[18] Hajo Broersma, Faustino Gomez, Julian Miller, Mike Petty, and Gunnar Tufte. Nascence project: nanoscale engineering for novel computation using evolution, jan 2012.

[19] K Clegg, J F Miller, K Massey, and M Petty. Travelling Salesman Problem solved in materio by evolved carbon nanotube device. In *Conf. on Parallel Problem Solving from Nature (PPSN)*, volume in Press of *Lecture Notes in Computer Science*. Springer, 2014.

[20] Kester Dean Clegg, Julian Francis Miller, Kieran Massey, and Mike Petty. Parallel Problem Solving from Nature – PPSN XIII: 13th International Conference, Ljubljana, Slovenia, September 13-17, 2014. Proceedings. chapter Travelling, pages 692–701. Springer International Publishing, Cham, 2014.

[21] Peter J Denning. Ubiquity Symposium 'What is Computation?': Editor's Introduction. *Ubiquity*, 2010(October), 2010.

[22] Ernesto Estrada. *The Structure of Complex Networks: Theory and Applications*. Oxford University Press, Inc., New York, NY, USA, 2011.

[23] Chrisantha Fernando and Sampsa Sojakka. *Pattern Recognition in a Bucket*, pages 588–597. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

[24] T C Fogarty, J F Miller, and P Thomson. Evolving Digital Logic Circuits on Xilinx 6000 Family FPGAs. In P K Chawdhry, R Roy, and R K Pant, editors, *Soft Computing in Engineering Design and Manufacturing*, pages 299–305. Springer London, London, 1998.

[25] T Freeth, Y Bitsakis, X Moussas, J H Seiradakis, A Tselikas, H Mangou, M Zafeiropoulou, R Hadland, D Bate, A Ramsey, M Allen, A Crawley, P Hockley, T Malzbender, D Gelb, W Ambrisco, and M G Edmunds. Decoding the ancient Greek astronomical calculator known as the Antikythera Mechanism. *Nature*, 444(7119):587–91, nov 2006.

[26] D E Goldberg. *GENETIC ALGORITHMS in search optimization & machine learning*. Addison Wesley, 1989.

[27] Jeffrey Goldstein. Emergence as a construct: History and issues. *Emergence*, 1(1):49–72, 1999.

[28] Garrison W Greenwood and Andrew M Tyrrell. *Introduction to evolvable hardware: a practical guide for designing self-adaptive systems*, volume 5. John Wiley & Sons, 2006.

[29] P.C. Haddow and G. Tufte. An evolvable hardware FPGA for adaptive hardware. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, volume 1, pages 553–560. IEEE, 2000.

[30] S L Harding and J F Miller. A Tone Discriminator in Liquid Crystal. In *Congress on Evolutionary Computation(CEC2004)*, pages 1800–1807. IEEE, 2004.

[31] Simon Harding and Julian F Miller. Evolution in materio: A real-time robot controller in liquid crystal. In *Evolvable Hardware, 2005. Proceedings. 2005 NASA/DoD Conference on*, pages 229–238. IEEE, 2005.

[32] Simon Harding and Julian F Miller. Evolution in materio: Evolving logic gates in liquid crystal. *Proc. Eur. Conf. Artif. Life (ECAL 2005), Workshop on Unconventional Computing: From cellular automata to wetware*, pages 133–149, 2005.

[33] J H Holland. *Adaption in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[34] John H. Holland. *Signals and Boundaries: Building Blocks for Complex Adaptive Systems*. MIT Press, 2012.

[35] John Henry Holland. *Hidden order: How adaptation builds complexity*. Basic Books, 1995.

[36] John Holmes. Discovering Risk of Disease with a Learning Classifier System. In *Proceedings of the 7th International Conference on Genetic Algorithms (ICGA97*, pages 426–433. Morgan Kaufmann, 1998.

[37] Philip Husbands and Owen Holland. The ratio club: A hub of British cybernetics. *The mechanical mind in history*, pages 91–148, 2008.

[38] Philip Husbands, Owen Holland, and Michael Wheeler. *The mechanical mind in history*. The MIT Press, 2008.

[39] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *European Conference on Artificial Life*, pages 704–720. Springer, 1995.

[40] Karl Young James P. Crutchfield. Computation at the onset of chaos. 1990.

[41] M W Johnson, M H S Amin, S Gildert, T Lanting, F Hamze, N Dickson, R Harris, A J Berkley, J Johansson, P Bunyk, E M Chapple, C Enderud, J P Hilton, K Karimi, E Ladizinsky, N Ladizinsky, T Oh, I Perminov, C Rich, M C Thom, E Tolkacheva, C J S Truncik, S Uchaikin, J Wang, B Wilson, and G Rose. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–8, may 2011.

[42] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse Ising model. *Phys. Rev. E*, 58(5):5355–5363, nov 1998.

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

[44] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. *arXiv preprint arXiv:1402.3511*, 2014.

[45] C G Langton. Computation at the edge of chaos: phase transitions and emergant computation. In S Forrest, editor, *Emergent Computation*, pages 12–37. MIT Press, 1991.

[46] Aristid Lindenmayer. Mathematical models for cellular interactions in development I. Filaments with one-sided inputs. *Journal of Theoretical Biology*, 18(3):280–299, 1968.

[47] Odd Rune Strømmen Lykkebø. Design and implementation of a prototype platform for evolution in materio. *NTNU*, 2010.

[48] B J MacLennan. A Review of Analog Computing. Technical Report UT-CS-07-601. Technical report, University of Tennessee, Knoxville, 2007.

[49] KIERAN MASSEY, MARK. Electrical Properties of Single-Walled Carbon Nanotube Networks Produced by Langmuir-Blodgett Deposition, dec 2013.

[50] M. K. Massey, A. Kotsialos, F. Qaiser, D. A. Zeze, C. Pearson, D. Volpati, L. Bowen, and M. C. Petty. Computing with carbon nanotubes: Optimization of threshold logic gates using disordered nanotube/polymer composites. *Journal of Applied Physics*, 117(13):134903, apr 2015.

[51] Mark K. Massey, Christopher Pearson, Dagou A. Zeze, Budhika G. Mendis, and Michael C. Petty. The electrical and optical properties of oriented Langmuir-Blodgett films of single-walled carbon nanotubes. *Carbon*, 49(7):2424–2430, jun 2011.

[52] M.K. Massey, D. Volpati, F. Qaiser, A. Kotsialos, C. Pearson, D.A. Zeze, and M.C. Petty. Alignment of liquid crystal/carbon nanotube dispersions for application in unconventional computing. In *PROCEEDINGS OF THE INTERNATIONAL CONFERENCE ON NUMERICAL ANALYSIS AND APPLIED MATHEMATICS 2014 (ICNAAM-2014)*, volume 1648, page 280009. AIP Publishing, mar 2015.

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

[54] Julian F Miller and Wolfgang Banzhaf. Evolving the program for a cell: from french flags to boolean circuits, 2003.

[55] J W Mills. The nature of the Extended Analog Computer. *Physica D: Nonlinear Phenomena*, 237(9):1235–1256, 2008.

[56] Jonathan Wayne Mills, M Gordon Beavers, and Charles A Daffinger. Lukasiewicz logic arrays. In *Multiple-Valued Logic, 1990., Proceedings of the Twentieth International Symposium on*, pages 4–10. IEEE, 1990.

[57] Melanie Mitchell, James P. Crutchfield, and Peter T. Hraber. Evolving cellular automata to perform computations: mechanisms and impediments. *Physica D: Nonlinear Phenomena*, 75(1-3):361–391, aug 1994.

[58] Maktuba Mohid, Julian F. Miller, Simon L. Harding, Gunnar Tufte, Odd Rune Lykkebo, Mark K. Massey, and Michael C. Petty. Evolution-in-materio: Solving function optimization problems using materials. In *2014 14th UK Workshop on Computational Intelligence (UKCI)*, pages 1–8. IEEE, sep 2014.

[59] Maktuba Mohid, Julian F. Miller, Simon L. Harding, Gunnar Tufte, Odd Rune Lykkebo, Mark K. Massey, and Michael C. Petty. Evolution-in-materio: Solving function optimization problems using materials. In *2014 14th UK Workshop on Computational Intelligence (UKCI)*, pages 1–8. IEEE, sep 2014.

[60] Maktuba Mohid, Julian Francis Miller, Simon L Harding, Gunnar Tufte, Odd Rune Lykkebo, Mark Kieran Massey, and Michael C Petty. Evolution-in-materio: Solving bin packing problems using materials. In *Evolvable Systems (ICES), 2014 IEEE International Conference on*, pages 38–45. IEEE, 2014.

[61] G Pask. Physical analougues to the growth of a concept. In *Mechanisation of Thought Processes*, number 10 in National Physical Laboratory Symposium, pages 877–922. Her Majesty's Stationery Office, London, UK, 1959.

[62] A. Pickering. Cybernetics and the Mangle: Ashby, Beer and Pask. *Social Studies of Science*, 32(3):413–437, jun 2002.

[63] Gheorghe Păun. Computing with Membranes. *Journal of Computer and System Sciences*, 61(1):108–143, 2000.

[64] L A Rubel. The Extended Analog Computer. *Advances in Applied Mathematics*, 14(1):39–50, 1993.

[65] Asaki Saito and Kunihiko K.b Kaneko. Inaccessibility and undecidability in computation, geometry, and dynamical systems. *Physica D: Nonlinear Phenomena*, 155(1-2):1–33, jul 2001.

[66] Hiroki Sayama. Introduction to the Modeling and Analysis of Complex Systems. aug 2015.

[67] Benjamin Schrauwen, David Verstraeten, and Jan Van Campenhout. An overview of reservoir computing: theory, applications and implementations. In *Proceedings of the 15th European Symposium on Artificial Neural Networks. p. 471-482 2007*, pages 471–482, 2007.

[68] CE E Shannon. Mathematical Theory of the Differential Analyzer. *Journal of Mathematics and Physics*, 20:337–354, 1941.

[69] Olivier Sigaud and Stewart W Wilson. Learning classifier systems: a survey. *Soft Computing*, 11(11):1065–1078, 2007.

[70] Petr Sosík. Universal computation with Watson-Crick D0L systems. *Theoretical Computer Science*, 289(1):485–501, 2002.

[71] Susan Stepney, Samuel L Braunstein, John A Clark, Andy Tyrrell, Andrew Adamatzky, Robert E Smith, Tom Addis, Colin Johnson, Jonathan Timmis, Peter Welch, Robin Milner, and Derek Partridge. Journeys in non-classical computation I: A grand challenge for computing research. *International Journal of Parallel, Emergent and Distributed Systems*, 20(1):5–19, 2005.

[72] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering*. Westview press, 2014.

[73] A Thompson, P Layzell, and R S Zebulum. Explorations in Design Space: Unconventional electronics design through artificial evolution. *IEEE Transactions on Evolutionary Computation*, 3(3):167–196, sep 1999.

[74] James Thomson. *Collected papers in physics and engineering*. Cambridge University Press, 2016.

[75] T. Toffoli. What Are Nature's "natural" Ways Of Computing? In *Workshop on Physics and Computation*, pages 5–9. IEEE, 1992.

[76] Tommaso Toffoli. Nothing Makes Sense in Computing Except in the Light of Evolution. *IJUC*, 1(1):3–29, 2004.

[77] Gunnar Tufte. Phenotypic, Developmental and Computational Resources: Scaling in Artificial Development. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, GECCO '08, pages 859–866, New York, NY, USA, 2008. ACM.

[78] Alan Mathison Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 237(641):37–72, 1952.

[79] D Volpati, M K Massey, D W Johnson, A Kotsialos, F Qaiser, C Pearson, K S Coleman, G Tiburzi, D A Zeze, and M C Petty. Exploring the alignment of carbon nanotubes dispersed in a liquid crystal matrix using coplanar electrodes. *Journal of Applied Physics*, 117(12):125303, 2015.

[80] John Von Neumann and Oskar Morgenstern. *Theory of games and economic behavior*. Princeton university press, 2007.

[81] Lee Wagner, Gunter; Altenberg. Complex Adaptations and the Evolution of Evolvability, 1996.

[82] James Alfred Walker, James A Hilder, and Andy M Tyrrell. Evolving Variability-Tolerant CMOS Designs. In Gregory S Hornby, Lukáš Sekanina, and Pauline C Haddow, editors, *Evolvable Systems: From Biology to Hardware: 8th International Conference, ICES 2008, Prague, Czech Republic, September 21-24, 2008. Proceedings*, pages 308–319. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[83] Norbert Wiener. *Cybernetics or Control and Communication in the Animal and the Machine*, volume 25. MIT press, 1961.

[84] S Wolfram. Universality and Complexity in Cellular Automata. *Physica D*, 10(1-2):1–35, 1984.

Chapter 6

The papers

The following are verbatim copies of submitted and reviewed papers, as described in Chapter 4.

6.1 Paper P1

Mecobo: A Hardware and Software Platform for In-Materio Evolution

Odd Rune Lykkebø, Simon Harding, Gunnar Tufte, and Julian F. Miller

Unconventional Computation and Natural Computation: 13th International Conference

Springer 2014

Mecobo: A Hardware and Software Platform for In Materio Evolution

Odd Rune Lykkebø[1], Simon Harding[2], Gunnar Tufte[1], and Julian F. Miller[2]

[1] The Norwegian University of Science and Technology
Department of Computer and Information Science
Sem Selandsvei 7-9, 7491 Trondheim, Norway
{odd.lykkebo,gunnar.tufte}@idi.ntnu.no
[2] Department of Electronics
University of York
Heslington, York, UK. YO10 5DD
slh@evolutioninmaterio.com, julian.miller@york.ac.uk

Abstract. Evolution in Materio (EIM) exploits properties of physical systems for computation. "Designs" are evolved instead of a traditional top down design approach. Computation is a product of the state(s) of the material and input data. Evolution manipulates physical processes by stimulating materials assessed in situ. A hardware-software platform designed for EIM experimentation is presented. The platform, with features designed especially for EIM, is described together with demonstration experiments using carbon nanotubes in a thick film placed on micro-electrode arrays.

1 Introduction

Unconventional computation and unconventional machines try to move beyond the Turing/von Neumann [7,11] concept of computing and computer architecture [9]. Evolution in Materio (EIM) [5,3,6] is such an unconventional approach where the underlying physical properties of bulk materials are explored and exploited for computation. In contrast to a traditional approach where a substrate, e.g. silicon, is meticulously designed, produced and programmed, the essence of EIM is neatly phrased as "bulk processes" producing "logic by the pound" by Stewart [8] when introducing his experimental electrochemical system.

Figure 1 illustrates a possible scenario for an EIM experimental set-up. The configurable material can be seen as a black box. Incident data are applied, the response is measured and evaluated against a predefined function. The search algorithm can manipulate physical properties of the material by applying configuration data vectors.

The format of input data, response and configuration data are material specific. As such, any experimental EIM set-up must be capable to produce configurations with properties capable of manipulating physical properties in the material. Incident data properties must be of a type capable of produce an observable response, i.e. output data.

O.H. Ibarra et al. (Eds.): UCNC 2014, LNCS 8553, pp. 267–279, 2014.
DOI: 10.1007/978-3-319-08123-6_22,

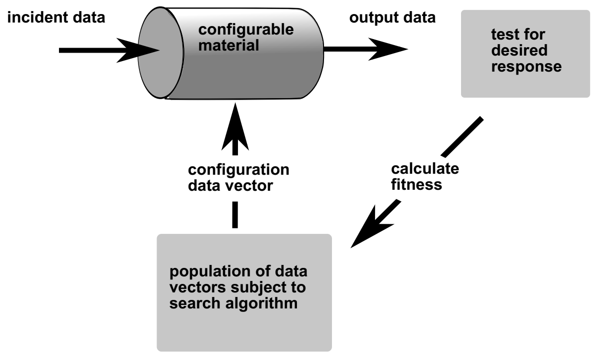


Fig. 1. Principle of evolution in materio

In Thompson's work [10] unconstrained evolution of configuration data for a Field Programmable Gate Array (FPGA) was used to evolve a tone discriminator. The FPGA may be considered as the material. The input signal to this digital circuit was analogue, the response was digital sampling of a captured analogue measurement. The configuration data for the chip was a digital bit stream. Even though Thompson exploited the physical properties of the chip, the configuration vector itself was digital. Harding and Miller [2] did a similar experiment with liquid crystal as material. In contrast to Thompson's experiment the configuration data signal property for the liquid crystal was evolved, the configuration data was unconstrained with regards to signal type, e.g. analogue, digital and time dependent.

In most EIM work an intrinsic approach has been taken, i.e. evaluation is performed on the physical material. An intrinsic approach allow access to all inherent physical properties of the material [6]. Intrinsic evolution requires an interface that can bridge the gap between the analogue physical world of materials and the digital world of EAs. We propose and demonstrate a flexible platform, Mecobo, designed to interface a large variety of materials. Flexible hardware allows for the possibility to map input, output and configuration terminals, signal properties and output monitoring capabilities in arbitrary ways. The platform's digital side, i.e. EA and software stack, is as important as the hardware. A flexible software platform including hardware drivers, support of multiple programming languages and a possibility to connect to hardware over the internet makes Mecobo a highly flexible platform for EIM experimentation.

Mecobo is part of a the NASCENCE project [1] targeting engineering of nano-scale units for computation. The demonstration experiments presented use single-walled carbon nanotubes mixed with poly(methyl methacrylate) (PMMA) dissolved in anisole (methoxy-benzene) as computational material.

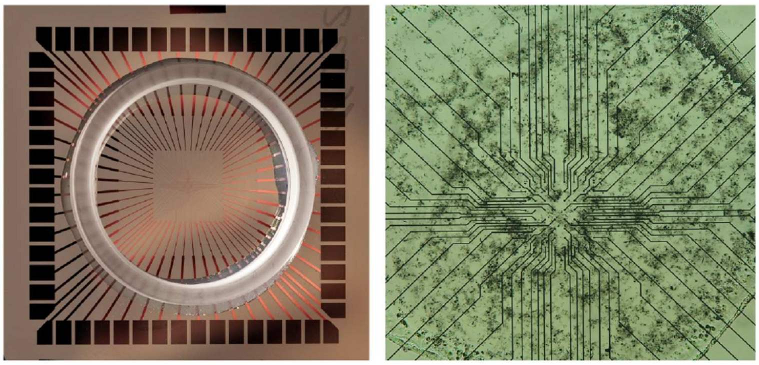
The article is laid out as follows: Section 2 presents the nanoscale material and the physical electrical terminals. In Section 3 the architecture of the hardware of the interface is presented. Section 4 presents the software of Mecobo. Experiments demonstrating the platform are presented in Section 5. Discussion and conclusions are given in Section 6.

2 Nano Material as Computational Resource

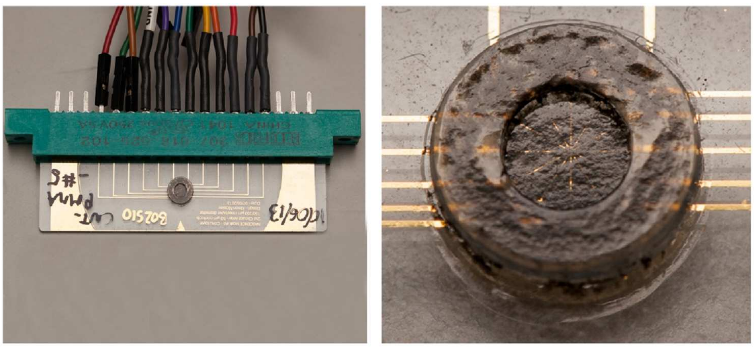
The demonstration experiments in Section 5 show computation in carbon nanotubes. The material samples used are all part of the ongoing NASCENCE project. At present time micro electrode arrays are used to connect electrically to a thick film containing nanotube structures.

Figure 2 show two different glass slides. In Figure 2(a) a slide with 64 electrodes is shown. Left; the glass slide with contacts on the rim. On the right a microscope image of the array covered with the thick film. A second micro electrode array is shown in Figure 2(b) the glass slide include a 12 electrode array. The micro electrode array slides was produced by Kieran Massey at the University of Durham by depositing a solution of carbon nanotubes onto a slide and letting the solvent dry out, leaving a random distribution of nanotubes across the probes in the micro electrode array.

In the demonstration of the interface in Section 5 the sample used was of the type shown in Figure 2(b). As the main topic here is hardware and software properties to interface a variety of materials, details regarding the physical properties are not presented in detail.



(a) Left: 64 electrode glass slide with contacts on the rim and electrode array in center. Right: close-up of the electrode array covered with a carbon nanotube thick film



(b) Left: 12 electrode glass slide with contacts on one side. Right: close-up of the electrode array covered with a carbon nanotube thick film

Fig. 2. Samples of materials placed on micro electrode arrays

3 Hardware: Interfacing the Black Box

Evolutionary exploration of computation by manipulation of physical systems is an intrinsic [10] approach. If the system is considered as a lump of matter, as illustrated in Figure 1, the selection of signal types, i.e. inputs, outputs and configuration data, assignment to I/O ports may not relate to material specific properties. As such, any I/O port can be assigned any signal type. Further, the signal properties, e.g. voltage/current levels, AC, DC, pulse or frequency, needed to unveil potential computational properties of different materials are unknown. To be able to explore and exploit a material's physical properties an experimental platform must have access to explore in as unconstrained a way as possible. However, in an evolutionary search the representation of genetic information, e.g. available voltage levels, will constrain the available search space.

3.1 Interface

The interface is designed to handle all the physical/electrical properties as mentioned above. To be able to ease the process of providing input data to any computational problem the interface also provides the possibility to provide input data. That is, a set of input data signals can be defined as part of the experimental set up to simulate external signals.

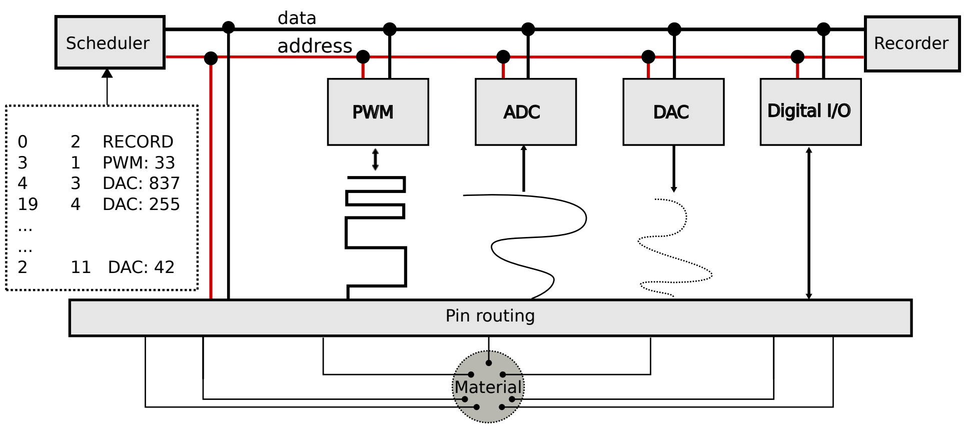


Fig. 3. Overview of the complete system

Figure 3 shows an overview of the hardware interface. In the figure an example set up is shown in the dotted box. The example genome defines pin 2 to be the output terminal, pin 1 to be the data input and pin 3 - 12 to be configuration signals. The architecture is controlled by a scheduler controlling the following modules: Digital I/O can output digital signals and sample responses. Analogue output signals can be produced by the DAC module. The DAC can be configured to output static voltages or any arbitrary time dependent waveform. Sampling of analogue waveforms from the material is performed by the ADC. Pulse Width Modulated (PWM) signals are produced by the PWM module.

The system's scheduler can set up the system to apply and sample signals statically or produce time scheduled configurations of stimuli/response. The recorder stores samples, digital discrete values, time dependent bit strings, sampled analogue discrete values or time dependent analogue waveforms. Note that the recorder can include any combination of these signals.

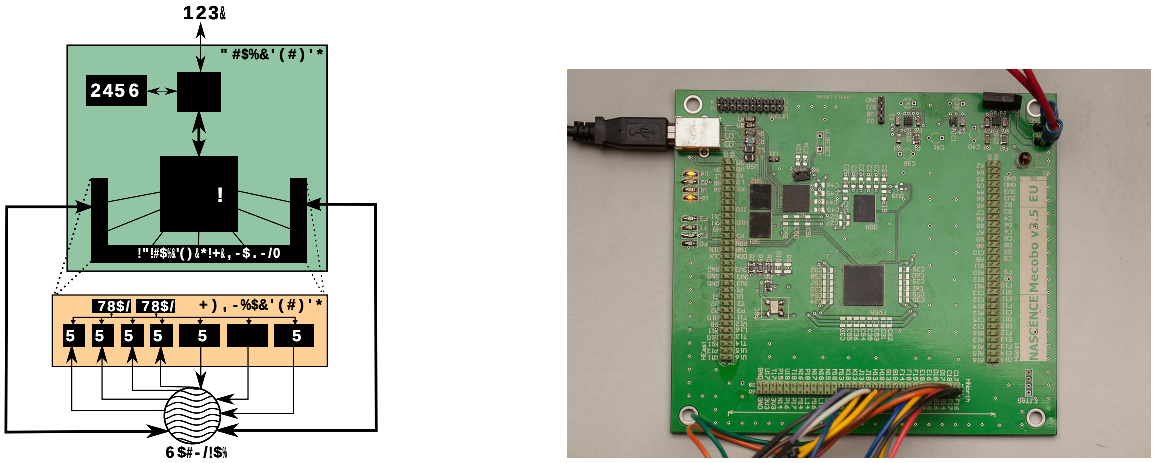
As stated, in a bulk materials there is no specific defined input and output locations, e.g. in the carbon nanotube PMMA samples is just distributed over the micro-electrode array. Thus it is desirable that the choice of data I/O and configuration terminals should be put under evolutionary control. In the interface all signals passes a crossbar, i.e. pin routing. Pin routing is placed between the signal generator modules and the sampling buffer (PWM, ADC, DAC, Digital I/O and Recorder) making it possible to configure any terminal of a material to be input, output or configuration.

The presented material signal interface in Figure 3 supports all our objectives. It is possible to evolve the I/O terminal placement. A large variety of configuration signals are available to support materials with different sensitivity, from static signals to time dependent analogue functions. The response from materials can be sampled as purely static digital signals, digital pulse trains or analogue signals. Further the scheduler can schedule time slots for different stimuli when time dependent functions are targeted or to compensate for configuration delay, i.e. when materials needs time to settle before a reliable computation can be observed.

3.2 Interface Physical Realization

The described system shown in Figure 3 is implemented as an autonomous interface hardware platform. The platform can communicate with a host computer over USB. The host can run an EA or stand as a bridge (server) connected to the internet.

The hardware implementation of the interface, which we call 'Mecobo', is shown as a block diagram in figure 4(a). Mecobo is designed as a PCB with an FPGA as the main component. The system shown in Figure 3 is part of the FPGA design together with communication modules interfacing a micro controller and shared memory. As shown in Figure 4(a) the digital and analogue designs are split into two. All analogue components are placed on a daughter board; such as crossbar switches and analogue-digital converters. This split enables redesign of the analogue part of the system without changing the digital part of the motherboard. The system shown in Figure 4(a) is an example of the current system, the Mecobo and an analogue daughter board. However, it is possible to include other extension boards to the Mecobo. The FPGA offers a possibility to include new modules adapted to any extension that can be connected to the digital I/O pin headers. The micro controller stands as a communication interface between the FPGA and the external USB port. The SRAM is available for the FPGA through the micro controller.



(a) Mecobo block diagram. (b) Picture of Mecobo.

Fig. 4. Hardware interface implementation overview

Figure 4(b) show the motherboard with the Xilinx LX45 FPGA, Silicon Labs ARM based EFM32GG990 micro controller connected to a 12 terminal material sample.

The motherboard is capable of controlling 80 digital I/O signals, which can be connected directly to a material sample or used for controlling resources on a daughter board. The FPGA drives the I/O pins at LVCMOS33 level, giving a minimum input voltage level for switching to digital 'high' at 2.0V, and maximum 0.8 corresponding to 'low'.

The software interface sends commands to the scheduling unit (implemented in the micro controller). The scheduler takes care of controlling the various pin controllers. A pin controller is the abstract term we use to describe a unit that drives or sources a physical I/O pin. Each pin controller has a slice of the global address space of this bus and can be programmed individually by the scheduling unit by outputting the command and data on the bus.

The scheduler accepts a sequence of commands from the user software. Each sequence item consists of parameters that describe the state of the pin at a given point in time. In Figure 3 for example, pin 2 is set as a 'recording' pin from time 0 (it also has a duration, and sampling frequency attached that is not shown here). Pin 1 is set to output a pulse width modulated version of the value 33, and pin 3 is set to output the analogue voltage level corresponding to 837, which could for instance map to analogue voltage level -2.3V relative to the daughter board analogue ground. In this case the scheduler would issue commands to one of the DAC controllers and to two of the PWM controllers.

4 Software

As there is no known, and hence no standard programming model for in-materio computation, we developed a system inspired by the track based model of music or video editing applications. An example of this is shown in Figure 5. Each

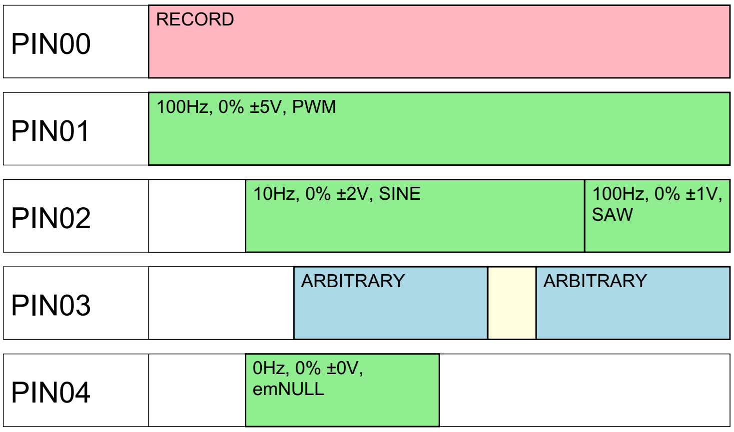


Fig. 5. Illustration of a genotype described in the 'track based' programming model. Each row is an output from the FPGA (and hence an input to the material). The horizontal axis is time. The model aligns closely with the hardware architecture, and also hints at a possible genotype representation.

track corresponds to an output pin of the FPGA, and on each track an action (or set of actions) are scheduled. Once the tracks are configured onto the FPGA, the sequence is 'played' back. As can be seen in the illustration, 'recordings' can also be scheduled. A recording in this case is the data captured from an input of the FPGA.

From this model, an Application Programming Interface (API) was developed that allows users to interact with the hardware. The main purpose of the API is to expose the functionality of the EM in a consistent and easy to use manner. Additional APIs provide support for data collection and for processing the data itself.

Client applications (i.e. software for performing the evolutionary algorithm), connects to a the EM via control software running on a PC. The control software is responsible for communicating at a low level to the EM, and translating the track based model into the FPGA's internal model.

The control software implements the API as a Thrift Server. Thrift[1] is a technology maintained by Apache that is designed to allow applications running on different operating systems, written in different languages and running on different computers to communicate with each other. Thrift provides a language that is used to define the functionality exposed by the server. This language is then compiled by the Thrift compiler into skeleton code that contains all the functionality needed to act as a server and accept connections, but is missing the functional components. These are then added to complete the server implementation.

On the client side, the interface can be compiled by Thrift into a library that exposes all the methods in the API. Thrift is able to generate the client and server code for many languages including C++, C#and Java. The client library

[1] http://thrift.apache.org/

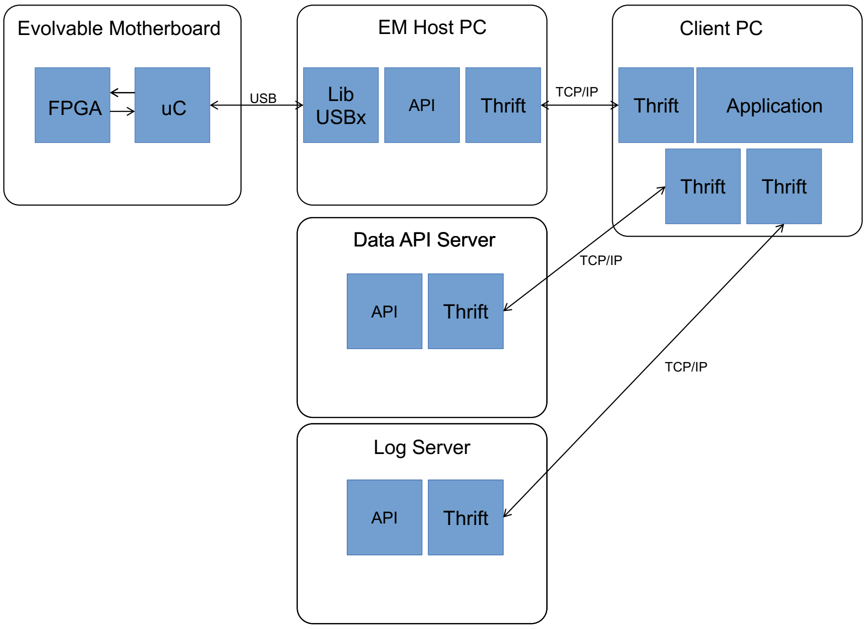


Fig. 6. Overview of the complete software architecture. Here the EA (user application) is run on a client PC. Communicating over TCP/IP to the EM host PC. The Mecobo platform is connected to and communicates with the host PC over USB. The log servers communicate with the client PC.

is then connected via TCP (or shared memory if the client/server are both on the same PC) to the server. Client applications then only need to implement their functionality, and no knowledge of the underlying protocols or workings of the EM or server software is required.

As the communication between Thrift Server and Client applications is based on TCP, there is no necessity for all components to run on the same computer. We have successfully tested the API over the internet, and have found that it is feasible for one institute to run the evolutionary algorithm, and another to host the EM.

Figure 6 shows the complete software architecture for the system. On the left we see the hardware (i.e. Mecobo), on the right is the client application. In the middle we see the main API components. Although only one EM is shown, it is possible to add more. Client applications can connect to multiple servers (and hence Mecobos), and hence can control a number of systems in parallel. We envisage this as being useful for robustness testing, investigating repeatability, and allowing multiple Mecobos to work together on a single problem.

5 Initial Experiments

To demonstrate the Mecobo platform, two experiments are presented. The experiments are executed on the presented hardware/software platform using a 12 electrode array similar to the shown example in Figure 2. The presented experiments only demonstrate a fraction of the capabilities of our hardware and software platform, and those that we expect the material to have. In the experiments an exhaustive search and a Genetic Algorithm (GA) was used. The exhaustive search was chosen to explore the potential functionality of the material at a coarse level. The GA approach is an example of how a search method can exploit the properties of materials toward achieving useful computation. However, other search/learning methods can also be used.

5.1 Exhaustive Sweep

If only digital time-independent logic is considered, then it is possible to run an exhaustive search mapping all possible configurations to the 12 pin sample. To take this into a more "computational" relevance, and to show the effect of interpreting the results when "programming" the material, we interpret two pins as input to a logic gates, the recording pin as gate output and the remaining 9 pins as configuration. This approach will leave 9 pins for configuration data. To represent functions a *gate output sum* can be constructed. The gate sum is the output of the truth table as shown in in table 1.

Table 1. Gate sum mapping. XOR. The gate sum is the decimal representation of the output column. 0110 binary give the gate sum 6.

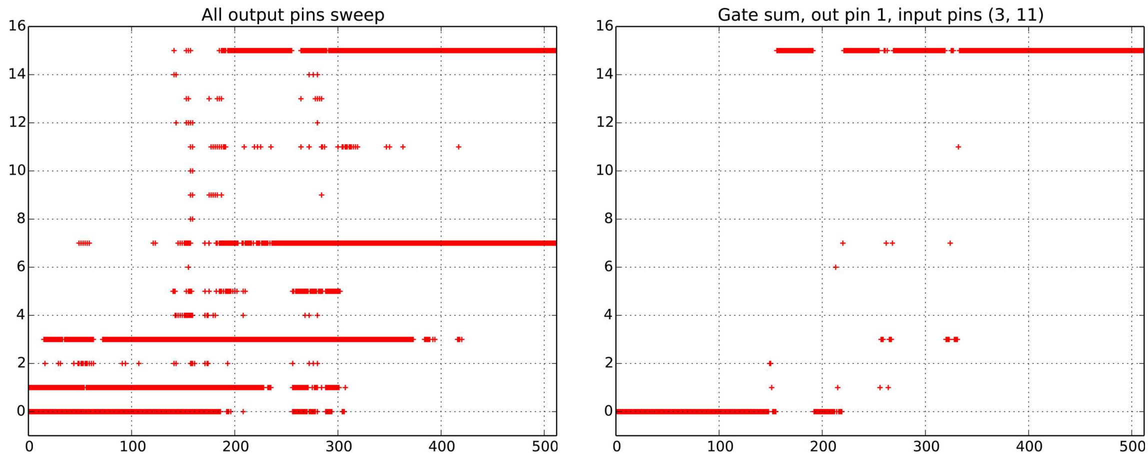
Input	Config	Output
0,0	1,0,1,1,0,1,0,1,1	0
0,1	1,0,1,1,0,1,0,1,1	1
1,0	1,0,1,1,0,1,0,1,1	1
1,1	1,0,1,1,0,1,0,1,1	0

A '1' represents 3.3V and '0' represents 0V. All possible pin combinations for input, configuration and output are tested and mapped to a functionality plot. If a gate is found it is plotted as a gate output sum, e.g. XOR: 0110 (6). An example of such a plot is presented in Figure 7(a). The plot show all found two input logical functions for all possible input output mappings.

The gate output sums are represented in decimal on the vertical axis. The configuration vector is given (decimal) on the horizontal axis. Interesting cases include XOR (gate sum 6 (0110)) and NAND (gate sum 8) are present together with all other possible 2 input logic function. Figure 7(b) show one of the possible I/O configuration. Here all possible gate configurations are shown for one particular I/O mapping, i.e. pin 1 output and pin 3 and 11 as input.

5.2 Genetic Search for Logic Functions

In Section 5.1 it was shown that the nanotube sample was capable of producing logic gates. However, to be able to evolve a desired functionality the EA must be able to exploit and explore the genetic representation and the search space, i.e. evolvability [4] must be present.



(a) Sweep all I/O combinations. The plot show all possible logical gates available in the material for all possible combinations of configuration and I/O pin mappings.

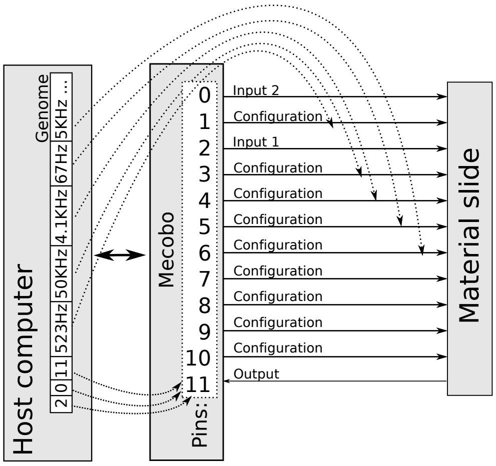
(b) Sweep fixed I/O combination. The plot show all possible logical gates available in the material if the configuration and I/O pin mapping is fixed.

Fig. 7. Example of logic gates found using exhaustive sweep. The x-axis uses a decimal representation of the 9 configuration bits. The resulting gate sum for each configuration is plotted. The decimal gate sum is given on the y-axis.

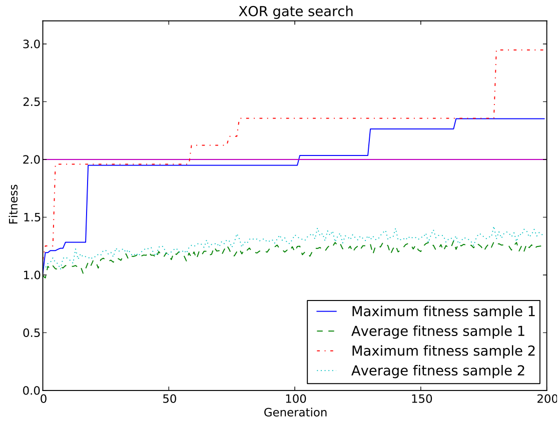
To explore evolvability, a Genetic Algorithm was used to search for stable XOR gates. The GA was quite standard; 25 individuals, two crossover points and tournament selection with 5 individuals as elite. Two different material samples was tested.

To demonstrate the platform's possibility to generate time dependent signals a genotype that allows dynamic PWM signals was chosen. The GA can adjust each configuration pin within a frequency range of 400Hz - 25MHz. The mapping of which material terminal to use as input, output or configuration was placed under genetic control. Figure 8(a) illustrates the representation. The first two genes assign the input signals, e.g. pin 0 and 2. The third gene gives the output, e.g. pin 11 is sampled by the recorder. The remaining 9 pins are mapped to configuration genes specifying frequency values. A restriction must be added to ensure that the same pin is not used for both input and output. To ensure "legal" gates, offspring with "illegal" pin mapping are not put back in the population. I/O and configuration genes were crossed over separately; hence the need for two point crossover.

The GA was set up to search for a stable 2-input XOR logic gate. The response is measured by setting up the gate input pins (i.e. the two first fields of the genome) to constant voltage levels at 0V or 3.3V. The configuration genes are frequencies of square waves, remaining fixed over the 2^2 possible inputs. The response is sampled from the GA-chosen output pin for 100ms at 10KHz. Referring to table 1, the number of correct samples in each sample period for each row is further multiplied by a constant, indicating how hard this case is to find: (0,0): 0.3, (1,0): 0.5, (0,1): 0.5, (1,1): 1.15. Particularly promising cases are



(a) Representation mapping from the EA genetic information to configuration and input/output signals connected to the material sample.



(b) Evolved XOR function in two different material samples.

Fig. 8. Experimental set-up. EA representation with unconstrained pin mapping and dynamic configuration signals are shown (left) with results for two different material samples (right).

further given a 0.5 bonus, giving a total possible fitness of 2.95 for a 'perfect' gate, and 1.95 for a functioning gate.

Figure 8(b) shows the evolution of fitness for these experiments. The horizontal line at 1.95 indicate the threshold for a functioning XOR gate where the majority of the samples in a sample buffer is over 55%. Note that a functioning XOR was found in both material samples, and in material sample 1 a near-perfect gate was discovered after 150 generations. The difference between the elite best and the average case is quite large. This can be explained by the relatively few XOR gates in the material, which can also be observed in figure 7(a).

6 Discussion and Conclusions

The presented hardware and software platform for EIM experimentation, known as Mecobo, is a tool enabling exploration and exploitation of materials for computation purposes. The flexibility in signal levels and types together with the possibility to put the mapping of input, output and configuration terminals under evolutionary control offers a possibility of relatively unconstrained material evolution.

The presented results demonstrate how the platform can be used. There are several interesting aspects worthy of note. In the exhaustive sweeps presented in Section 5.1 the gate sum plots are a course mapping of possible computational properties of the material. The plot in Figure 7 show that this sample is capable of solving problems beyond simple threshold functions. As such, results for such exhaustive sweeps can be used to coarsely classify a material and be used to measure the closeness/distance between materials samples within a batch or batches with different physical properties.

Even if a material is capable of implementing a function, like the XOR, it is not necessarily easy to evolve. Further, as indicated by earlier EIM work, the stability of discovered solutions may be problematic. The example given shows two important factors. Stability can (and should be part) of the problem definition. The change of representation shows the possibility to provide a variety of signal types. The material used in the example show computational properties for static and dynamic configuration data.

As stated in Section 1 this work is part of a bigger project. The platform is in use by several researchers in the NASCENCE project consortium, e.g. University of York for function optimization and machine Learning classification problems.

Software, HDL code, schematics and PCB production files for the platform can be downloaded from the *download resources* at: *http://www.nascence.eu*.

Acknowledgements. The research leading to these results has received funding from the [European Community's] Seventh Framework Programme ([FP7/2007-2013] [FP7/2007-2011]) under grant agreement no [317662]. We are grateful to Kieran Massey and Mike Petty for the preparation of materials and the microelectrode array.

References

1. Broersma, H., Gomez, F., Miller, J.F., Petty, M., Tufte, G.: Nascence project: Nanoscale engineering for novel computation using evolution. International Journal of Unconventional Computing 8(4), 313–317 (2012)
2. Harding, S.L., Miller, J.F.: A tone discriminator in liquid crystal. In: Congress on Evolutionary Computation(CEC2004), pp. 1800–1807. IEEE (2004)
3. Harding, S.L., Miller, J.F., Rietman, E.: Evolution in materio: Exploiting the physics of materials for computing. Journal of Unconventional Computing 3, 155–194 (2008)
4. Kirschner, M., Gerhart, J.: Evolvability. Proceedings of the National Academy of Sciences of the United States of America 95(15), 8420–8427 (1998)
5. Miller, J.F., Downing, K.: Evolution in materio: Looking beyond the silicon box. In: 2002 NASA/DOD Conference on Evolvable Hardware, pp. 167–176. IEEE Computer Society Press (2002)
6. Miller, J.F., Harding, S.L., Tufte, G.: Evolution-in-materio: Evolving computation in materials. Evolutionary Intelligence 7(1), 49–67 (2014), `http://dx.doi.org/10.1007/s12065-014-0106-6`
7. von Neumann, J.: First draft of a report on the EDVAC. M. D. Godfrey (ed.) (1992), Technical report. Moore School of Electrical Engineering University of Pennsylvania (1945)
8. Stewart, R.M.: Electrochemically active field-trainable pattern recognition systems. IEEE Transactions on Systems Science and Cybernetics 5(3), 230–237 (1969)
9. Teuscher, C., Adamatzky, A.: Unconventional Computing 2005: From Cellular Automata to Wetware. Luniver Press (2005)

10. Thompson, A.: An evolved circuit, intrinsic in silicon, entwined with physics. In: Higuchi, T., Iwata, M., Weixin, L. (eds.) ICES 1996. LNCS, vol. 1259, pp. 390–405. Springer, Heidelberg (1997)
11. Turing, A.M.: On computable numbers, with an application to the Entscheidungsproblem. In: Proceedings of the London Mathematical Society 1936-1937. ser. 2, vol. 42, pp. 230–265. Mathematical Society, London (1937)

6.2 Paper P2

Comparison and Evaluation of Signal Representations for a Carbon Nanotube Computational Device

Odd Rune Lykkebø and Gunnar Tufte

IEEE International Conference on Evolvable Systems 2014

IEEE 2014

Comparison and Evaluation of Signal Representations for a Carbon Nanotube Computational Device

Odd Rune Lykkebø
The Norwegian University of Science and Technology
Department of Computer and Information Science
Sem Selandsvei 7-9, 7491 Trondheim, Norway
Email: odd.lykkebo@idi.ntnu.no

Gunnar Tufte
The Norwegian University of Science and Technology
Department of Computer and Information Science
Sem Selandsvei 7-9, 7491 Trondheim, Norway
Email: gunnart@idi.ntnu.no

***Abstract*—Evolution in Materio (EIM) exploits properties of physical systems for computation. Evolution manipulates physical processes by stimulating materials by applying some sort of configuration signal. For materials such as liquid crystal and carbon nanotubes the properties of configuration data is rather open. In this work we investigate what kind of configuration data that most likely will be favourable for a carbon nanotube based device. An experimental approach targeting graph colouring is used to test three different types of signal representation: static voltages, square waves and a mixed signal representation. The results show that all signal representation was capable of producing a working device. In the experiments square wave representation produced the best result.**

I. INTRODUCTION

The enormous success and impact of today's computers can be traced to the success of the human designer's intent and ability to follow Moore's law to create ever smaller basic computational building blocks and the scalability of the von Neumann architecture [7].

A lot of work in the world of unconventional computation has been focused on the observation that we are fast approaching the physical limits of silicon based transistors. At some point we will reach a point where the building blocks are made of single atoms [1] making physical manipulation harder, the so-called 'power wall' is fast approaching, etc.

There are several computing paradigms that seek to solve these problems: chemical computation[2], natural computation[3], DNA computing[4], and a returning problem is how to program these novel systems. Some attempts have been made to create more conventional programming languages on top of such systems, e.g. Amorphous computing[5], or as is the case with conventional computers operating under the Turing model using von Neumann-hardware[6], [7], a stream of instructions.

However, in [8], Stepney writes "we should first investigate these systems to discover what computationally-interesting processes they perform "naturally": we should be physical process-driven rather than abstract model-driven". This observation shifts the focus from trying to coerce a physical system to follow established rules about what a computation is, to simply attempt to exploit the *natural* capabilities of the physical system to do computation. These ideas are not new; Gordon Pask's ear is an example of such hardware-before-software thinking[9].

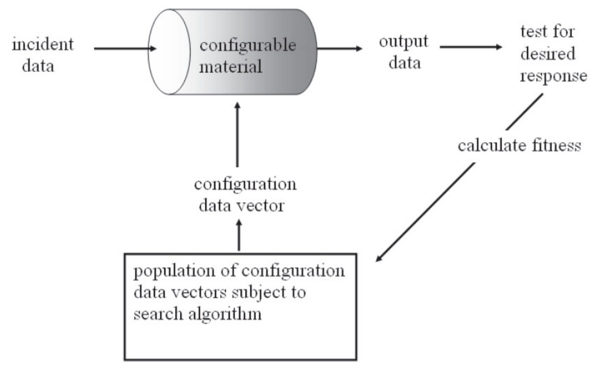


Fig. 1: Principle of evolution in materio. A configurable material is manipulated by applying a configuration data vectors. The search algorithm manipulates physical properties of the material toward a predefined computational function.

Evolution-In-Materio (EIM) [10], [11] aim at manipulating underlying physical properties of materials for computation. The target material for our work is based on Carbon NanoTubes (CNTs).

The program in EIM is replaced by manipulation of physical properties and process in a material as to achieve computation. The manipulation can, as in this paper, be electrical signals that can set the material in a state that can achieve the desired computation. However, what class and range of signals to manipulate materials toward computation is not given.

Figure 1 illustrates the basic concept of an EIM set-up. A material, i.e. a configurable material, is exposed to a configuration vector that manipulates the materials internal physical properties to compute. The computation can be a transfer function taking the incident data as input and producing the result as output data. The configurable material can be seen as a black box. The configuration data vector for a given computation is unknown. In the example shown an Evolutionary Algorithm (EA) is used to search toward a configuration vector that can manipulate the physical properties of the material to a state that can perform the desired computation.

The format of input data, response and configuration data are material specific. As such, any experimental EIM setup must be capable of producing configurations with properties that can manipulate physical properties in the material and the incident data properties must be of a type capable of produce an observable response, i.e. output data.

In Thompson's experiments, e.g. tone discriminator [12], Field Programmable Gate Arrays (FPGAs) was the target device. In these experiments the manipulation of the device was done by unconstrained evolution [13] of the configuration bits configuring the FPGA. Harrding and Miller used Liquid Crystal in the form of a LCD as a target device [14]. In these experiments the manipulation of the material was analogue voltage levels, i.e. a number of analogue voltage levels was used to configure, or program, the LCD device. In experiments using CNTs analogue voltage levels [15], digital bit vectors [16] and pulse trains [17] has been used with success.

For such devices as the LCD- and the CNT-based devices the type of which signals to use for manipulation is not given. However, some basic requirements do exist; the manipulation signals must be capable of producing change in the target material, i.e. manipulating the underlying physical properties. Furthermore, the input signals (data) must be in a range that can be sensed by the physical structure of the material and the output (result of computation) must be observable from the outside.

Another challenge of EIM, and most related approaches is the definition of the border between the apparatus applying configuration signals, and the material itself. As shown in the work of Thompsons, Layzell and Zebulum temperature and/or electrical noise in the laboratory influence and can even contribute as an exploited parameter as part of the evolved configuration [18]. Such influences can make it hard to define what information that are present in the genotype and the phenotype. In an setup similar to the illustration in Figure 1 the genotype is the applied configuration vector. We may define the vector as a bit string.

However, in the material there is no definition of a bit, the material is stimulated by a voltage resulting in a current. As such, the current is a product of the impedance of the material and the voltage source. That is: there is a mapping that we may not control or even be aware of.

To address the two challenges we investigate the evolutionary response of a CNT based device to different types of configuration signals. Such an exploration will give us an insight in how to design genotypes that are adapted to such a device. further, the method can be applied to any material as to map the material's underlying physical properties sensitivity to be manipulated. As to set a border between the genotype, phenotype and the used experimental equipment an extended phenotype is proposed. Such an extended phenotype will limit the unknown factors of the environment.

The material used and the experimental platform is briefly presented together with a discussion on the mapping challenge in Section II. Section III presents the experimental approach to explore different configuration signals to the CNT example

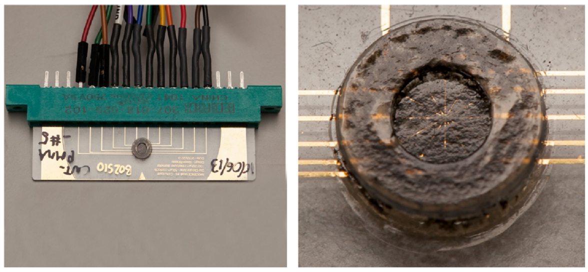


Fig. 2: 12 microelectrode array on a glass slide. The slide is placed in a sample holder (left) and a close-up of the CNT device (right).

material. Section IV presents the results from running the experiments and a discussion on the findings. Section V discusses the findings in the context of Evolution-in-Materio and further work.

II. Interfacing Materials: Representation and Mapping

As stated in Section I an EIM approach is based on the possibility to evolve a configuration that influence the physical properties of materials toward doing computation. Such a system require a material that can be manipulated, some apparatus that can stimulate the material and an EA with a representation that can explore and exploit the material toward achieving the computational function sought.

A. *Carbon Nanotubes as a Computational Device*

The target material of this work is carbon nanotubes placed on a microelectrode arrays to connect electrically to a thick film containing nanotube structures. Figure 2 show a material sample glass slide in a sample holder and a close-up of the actual CNT sample on the microelectrode array. The shown sample has 12 electrodes that can be stimulated by applying electrical signals on a set of electrodes, i.e. the configuration vector; however our experiments use a 16-electrode version.

The material, i.e. computational device, is made of single-walled carbon nanotubes mixed with poly(methyl methacrylate) (PMMA) dissolved in anisole (methoxy-benzene). Samples are produced by by depositing a solution of carbon nanotubes onto a slide and letting the solvent dry out, leaving a random distribution of nanotubes across the probes on the micro electrode array.

Devices such as the previous described LCD [14] and the CNT devices used here is not produced with specific signals, e.g. microelectrodes, representing data inputs, data outputs or configuration data. As such, any microelectrode can be assigned to be any signal type.

The micro electrode array and the CNT computational device was produced by Kieran Massey at the University of Durham as part of the NASCENCE project [19]. Information on the physical parameters and electrical characterization of the devices are described in [20].

B. Interfacing Carbon Nanotubes

In most EIM work an intrinsic approach has been taken, i.e. evaluation is performed on the physical material. An intrinsic approach allow access to all inherent physical properties of the material [11]. Intrinsic evolution requires an interface that can bridge the gap between the analogue physical world of materials and the digital world of EAs.

The CNT devices used herein is interfaced to an EA by a custom made hardware interface. The interface is named Mecobo. The Mecobo hardware and software is presented in detail in [17]. A PC running an EA can map any microelectrode to be input data, output data or configuration data, i.e. port mapping can be under genetic control. Mecobo open for applying a wide range of data and configuration signals, e.g. voltage levels, AC, DC, pulse trains or frequency variation.

The Mecobo platform is shown in Figure 3. Figure 3(a) show the overall design. Configuration specification are loaded from a PC to Mecobo over the USB port. The microcontroller communicates with the USB interface and with an FPGA on an internal bus. The FPGA can interface directly to materials or as in the figure use a daughterboard to extend the signal range.

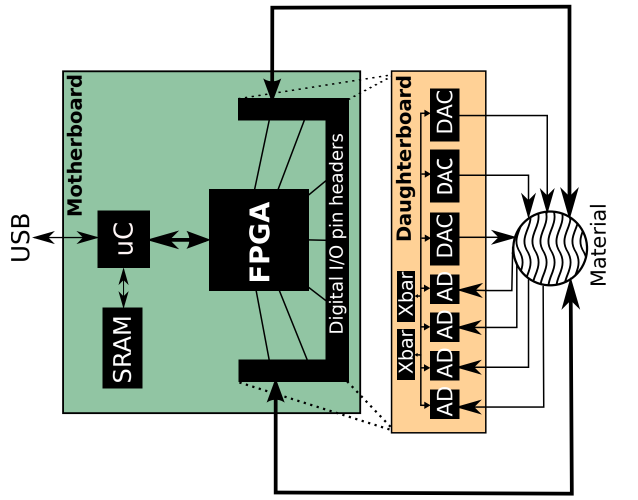
Mecobo is shown at a functional level in Figure 3(b). The software interface sends commands (over USB) to the scheduling unit (implemented in the micro controller). The scheduler takes care of controlling the various pin controllers. A pin controller is the abstract term we use to describe a unit that drives or sources a physical I/O pin. Each pin controller has a slice of the global address space of this bus and can be programmed individually by the scheduling unit by outputting the command and data on the bus. The scheduler accepts a sequence of commands from the user software. Each sequence item consists of parameters that describe the state of the pin at a given point in time. In Figure 3(b) for example, pin 2 is set as a recording pin from time 0.

In Figure 3(c) we see a Mecobo motherboard with a mixed signal daughterboard attached on top. A CNT glass slide is plugged into the daughterboard. The flexibility of Mecobo allow the necessary freedom of pin routing and type of configuration signal necessary for investigating different configuration signal types as to achieve computation in materials with the flexibility of CNT based materials.

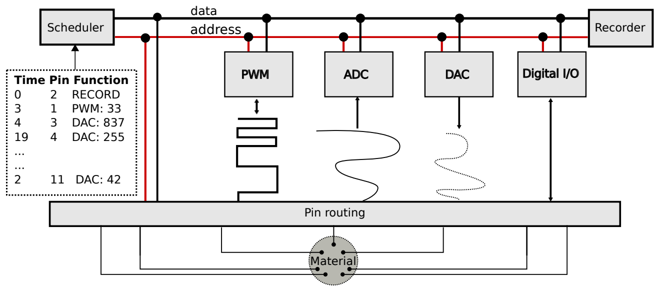
C. EAs, representation and mapping

A Genetic Algorithm (GA) operates on a genome, representing a potential solution to a problem, usually referred to as the phenotype. In this paper, we are attempting to build computational device from an unorganized sample of material by manipulating it using electric currents at various voltage levels. Voltage is a physical property of current, and through our Evolution-in-Materio platform Mecobo we can choose to input current with voltages of various *signal forms*, i.e. square waves with varying duty cycles, voltage amplitudes, and so on.

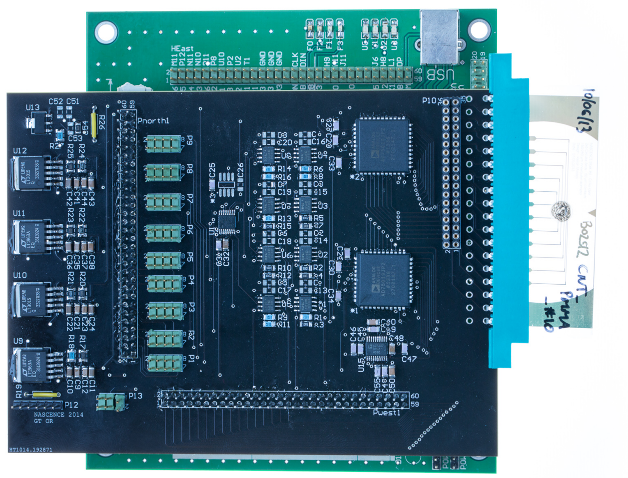
Since we must measure the fitness of an actual physical entity, it is clear that the output from the material is also a fundamental part of what the GA is building. In a sense, we



(a) Block diagram of the Mecobo hardware interface.



(b) Mecobo architecture with functional units.



(c) Picture of the Mecobo motherboard with mixed signal daughterboard.

Fig. 3: Overview of the Mecobo hardware interface.

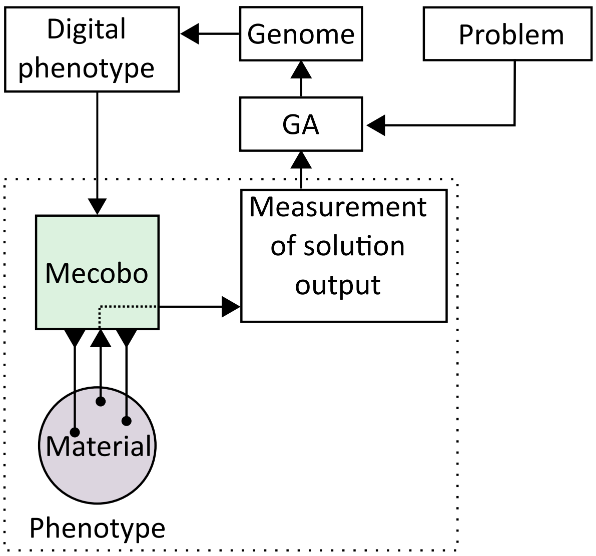


Fig. 4: One of the ways to label the system in the context of GA's. The problem is encoded in the fitness function of the GA.

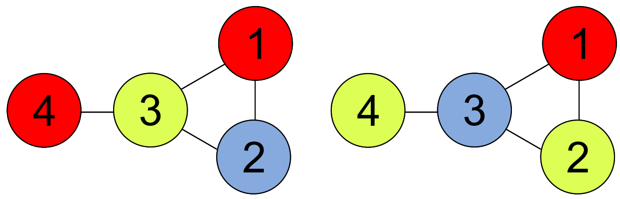


Fig. 5: Two valid 3-colourings of the same graph. It is a solution because there are no two neighbours with the same colour. The numbers are node labels, which are mapped to outputs from the device we seek to build.

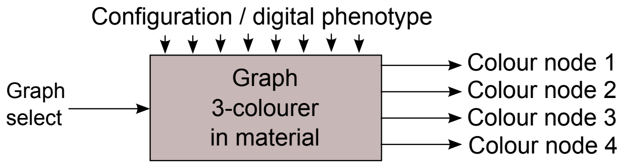


Fig. 6: The device we seek to build. The graph select input is used to select which of the two possible colourings shown in figure5 should be output.

are not only evolving a solution to a problem, but also the ability for us to capture that solution.

To be able to solve a problem, we have to express the problem instance with a genotype which the GA uses it's operators on. In a standard EA approach, the EA generates genotypes using some representation that maps directly to a candidate solution, a phenotype, via the genotype-phenotype map. This phenotype can be input into the fitness function and evaluated. One can choose to look at this set of signal forms as a phenotype, and the fitness evaluation to apply the phenotype on the material. This is shown in Figure 4, where we have called this the digital phenotype, for reasons we will explain.

Due to the above mentioned fact that we are building a physical entity whose output directly affects the fitness measurement, it seems more apt to consider Mecobo and the material the 'real' phenotype; and the output from the material (or rather, the sampled output collected by the Mecobo platform) to be the candidate solution that is input into the fitness function of the GA.

In this case, the genotype-phenotype map is further extended to include the entire individual being evaluated by the GA. To ease the discussion forward, we will call the set of signal forms the *digital phenotype*, referring then explicitly to the waveforms we schedule to be applied to the material. It is worth noting that this is just one way of observing the system–going into the philosophical debate about what constitutes the phenotype and what is part of the environment is beyond the scope of this paper but is worth exploring in further work.

Rather, in this paper we want to investigate how the choice of signal form in the genotype-digital phenotype affects the performance of the GA, under a 'direct mapping'. By direct mapping we mean that the GA operates directly on the signal forms, our representation, manipulating it by directly changing parameters of the signal forms, such as frequencies of square waves and voltage levels. We will do this by choosing a problem whose solution we encode in a fitness function. We will then attempt to evolve a device that solves this problem using three different constraints.

III. The Experiment

As explained in the introduction, Mecobo can produce a number of different signal forms, which can be used by a Genetic Algorithm (GA) to build novel ways of configuring the material under exploration, so as to produce a *computational device* in some sense.

In this experiment we will investigate the effect of limiting the signal forms that the GA can use to build a particular sought-after device. The device which we will build is one that is capable of producing two valid 3-colourings of a simple graph with 4 nodes. Two such colourings are shown in figure 5.

The problem is formulated as follows: Given 3 different colours, colour a graph such that no two neighbouring nodes of the graph are assigned the same colour. For more details of this problem in the context of Genetic Algorithms, see [21].

We will use a GA to construct (or find) a device that produces such colourings in the material connected to Mecobo.

The conceptual device is shown in figure 6. It consists of one 'problem select' input signal, 8 configuration signals and 4 output signals.

We *define* a colouring of one node as the highest voltage seen on an output pin, which is assigned to a particular node. For instance, the colour 'red' in figure 5 appears on node 1 and 4 in the first graph and 1 in the second graph. These nodes are assigned to output pins and we colour a graph if it is possible to separate these colour measurements. As such, the problem is encoded implicitly in the fitness function. We are currently only investigating the actual output from the device, not *how* it achieves the output.

To say anything meaningful about the effect of limiting signal forms, we will measure and compare how quickly (in number of generations) the GA can find a working solution, and how high fitness this solution achieves. This allows us to more quickly focus the effort of tuning the GA to the material under test, *and* which problem we are attempting to solve using the material. Since the search room of potential ways of exciting the material is very large (essentially any real physical phenomena can be used), it is in our interest to find ways of ruling out certain parts of it.

We will also measure the average fitness of the population to investigate if there is a connection between highly fit individuals and the average population fitness, or if highly fit individuals are mere 'artefacts' that pop up randomly over the course of the evolutionary run.

A. Genome and population initialization

Each device corresponds to one particular individual in the GA. The individual's genome consists of a list of 13 *sequence items*. One item is used for the input pin, 8 for the configuration data and 4 for recording. Each such sequence item have some the following fields defined, depending on the signal form and if the pin is a 'recording' pin.

- Physical pin, p
- Signal form, s
- Time to start, t_s
- Time to end, t_e
- Amplitude of voltage, a
- Frequency, f,
- Duty cycle, d
- Recording rate, r

The signal form field s consists of one the following options:

1) Square waves: Only using frequencies and duty cycles of square waves, keeping the amplitude of the waves fixed at 0 and 3.3V.
2) Static: Only using static voltage levels from 0 to 3.3V.

In each individual, for all 8 configuration sequence items, the choice of s field is limited to the allowed type, i.e. only digital square waves, only analogue voltage levels or a mix of both. t_s and t_e is kept fixed at 0 and 100, respectively. a is only set when s is of analogue output type, and similarly f and d is only set when s is set to square wave output. 4 of the individuals is further designated as recording pins, one per node in the graph, with fixed recording rates r of 10 KHz.

The GA manipulates each of these sequence items directly, and as mentioned in 4, this means that the genotype-(digital) phenotype map is direct. The input is either 0 or 3.3V (we have two different graphs so we need only two different inputs), and it is replicated across all of the 4 the remaining pins.

B. GA details

Selection is done using a tournament selection with a selection pressure of 0.8. The breeding is done using two uniformly random selected parents. A single point crossover is employed to create two new offspring. A cross-over pin k is chosen from a uniform random distribution, and the two offspring are created by concatenating sequence items belonging $0..k$ from parent 1, and $k+1$..n from parent 2. For offspring 2, the role of the parents is reversed.

For mutation, each gene in the offspring (i.e. a sequence item) has probability 0.92 to be mutated. There are two mutation operators, one that moves the physical pin either up or down, and one in which a quantity drawn from a Gaussian distribution is added to the frequency and duty cycle in the case of a square wave signal form, and to the voltage amplitude in the case of a static voltage level.

The GA is run for 100 generations, with a population size of 65.

C. Fitness measurement

As mentioned, in each tested individual there is a mapping from one node to one output pin from the material. Hence for a 4 node problem, there will be 4 output measurements, which is done at 10KHz and collected into 4 sample buffers.

We then take the *highest* seen voltage in one such buffer to represent the colour of each node, 1, 2, 3 and 4, as seen in figure 5. Each individual is then applied twice to the material through the Mecobo board; one time with all the input pins set to 1, and one with the input pins set to 0.

The key definition we use to separate the colors is thus $Red(R) > Green(G) > Blue(B)$.

The voltage measurements (or colours) are collected into 3 lists, c_f, where f are the 3 possible colours; R, G, B. For the graph in 5 this means that c_R will be a list of 3 measurements (colours), c_G also contain 3 and c_B has 2.

To be a working device, all of the voltage measurements in c_R has to be higher than all of the measurements in c_G, and all measurements in c_B must he higher than c_B.

We therefore define the fitness function as follows.

$$\begin{aligned} fitness = k * ((minc_R - maxc_G) \\ +(minc_G - maxc_B) \\ -(maxc_G - minc_G)) \end{aligned} \tag{1}$$

Where min and max means the smallest and largest element in the measurement list, respectively.

In other words, we seek to make the distance between the lowest 'red' measurement and the highest 'green' measurement as big as possible, allowing small increments in the fitness. We also seek to minimize the difference between the biggest and smallest green measurement, so as to separate from red and blue. Note that by our definition of a working device, a device can have relatively low fitness and still be "working"– we therefore also check each individual, and give them a slight bonus ($k = 1.2$) if we find them to color the graphs correctly. We could give all individuals that are found to be not working a very low score; but we found that doing this quickly limited the diversity of the populations and a non-working local maxima was soon found. Increasing fitness means that there is a higher chance of an individual to show the wanted behaviour sought.

IV. Results

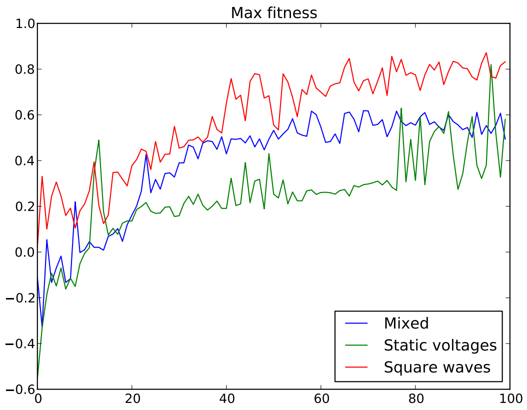


Fig. 7: Three evolutionary runs capped at 100 generations.

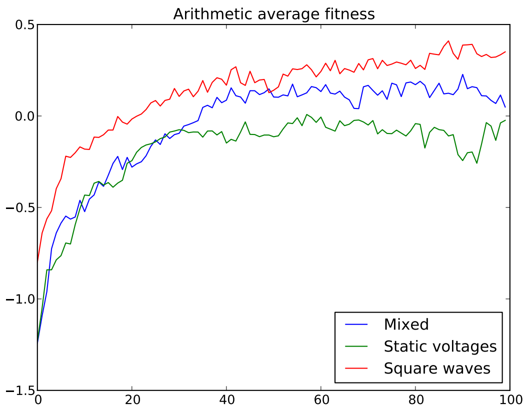


Fig. 8: Arithmetic mean of three evolutionary runs capped at 100 generations.

Each of the signal forms produced a working device per the definition presented in section III. However,to be a working device, the output from the device must produce a 'switching' behaviour. By this it is meant that an output pin assigned to node 3, has to switch from one colour to another with the only difference being the input pins being either high or low. The same goes for node 2, as seen in figure 5.

The most successful signal form for this particular fitness function was square waves, producing a total of 11 working devices during the course of the run, most devices appearing in generations 90 to 100. This trend continues in supplementary experiments we have run with more than 1000 generations.

Using static analogue voltages we obtain only 1 working device during the 100 generations run. This occurs at generation 64, when the mean fitness have stabilized, see figure 8. The first working device for square waves appears at around generation 50, where we see a similar overall flattening of the fitness in figure 8. For the mixed signals the first working device appears at generation 65, after a similar flattening. When analysing the logs from the runs, we find that by around generation 50 the diversity in terms of pin positions have lessened– the crossover operator has presumably found a range of pins that show more interesting behaviour than the others.

When comparing the most fit individuals in table I this is also evident. It seems plausible that the GA has found an interesting region in the fitness landscape and keeps trying to optimize the behaviour found here to configure a working device. The best working square wave individual achieve a voltage separation of over 0.15V between 'red' and 'green', and 0.13V for the mixed case. This is well within the noise margins of the AD converters, and gives us an indication about what voltage levels we must be able to observe in further work. For the 'green' and 'blue' separation, the findings are similar.

Furthermore, comparing figure 8 and 7 support the idea that it is healthy for the population to have high fitness when searching for this type of device in the material.

V. Conclusion

We must be careful about drawing too many hard conclusions from these results. As we have argued before, the choice of signal form is highly dependent on the fitness function, and what type of computation we are trying to achieve. In spite of this, for this particular fitness function in which we wish to see a 'switching' characteristic of a certain output pin, square waves seem outperform the two other candidates.

More generally several observations was made: The fact that each of the signal forms produced a working device indicate that the CNT samples used is indeed evolvable. However, the difference also indicate that the representation of signals in the configuration vector influence on the evolutionary result. In the presented experiments all signal types produced working devices but as stated the experiment was set-up to be quite easy solvable as to be able to detect differences. If more complex/difficult problems are to be targeted the results show that the choice of representation of signals clearly influence the results, implying they most likely also influence which problems that can be targeted.

The good results for square waves are promising. Square waves are far easier to generate then analogue static voltages and mixed signal stimuli when it comes to drawing a border for genotype, phenotype and mappings, because they can be output directly from the FPGA without going through DA-conversion.

Acknowledgment

The research leading to these results has received funding from the [European Community's] Seventh Framework Programme ([FP7/2007-2013] [FP7/2007-2011]) under grant agreement no [317662].

We are grateful to Kieran Massey and Mike Petty for the preparation of materials and the microelectrode array. Thanks to Simon Harrding for Mecobo interface software.

Pin(s)	Form	Params
9	Out 1	
4	Out 2	
14	Out 3	
15	Out 4	
1,3,7,11	In	
0	Analog	2.7V
2	Analog	3.0V
5	Analog	3.1V
6	Analog	0.8V
8	Analog	3.3
10	Analog	3.3V
12	Analog	3.2V
13	Analog	0.0V

Static analog

Pin(s)	Form	Params
6	Out 1	
3	Out 2	
14	Out 3	
13	Out 4	
8,9,10,11	In	
0	Sq.Wave	1MHz, 28%
1	Sq.Wave	252KHz, 93%
2	Sq.Wave	893KHz, 45%
4	Sq.Wave	657KHz, 50%
5	Sq.Wave	656KHz, 81%
7	Sq.Wave	898KHz, 33%
12	Sq.Wave	731KHz, 47%
15	Sq.Wave	25KHz, 0%

Square waves

Pin(s)	Form	Params
6	Out 1	
4	Out 2	
12	Out 3	
11	Out 4	
0,1,2,15	In	
3	Sq.Wave	497Kz, 100%
5	Sq.Wave	322KHz, 100%
7	Analog	3.1V
8	Analog	1.4V
9	Sq.Wave	37KHz, 10%
10	Sq.Wave	328KHz, 30%
13	Analog	0.2V
14	Sq.Wave	261KHz, 37%

Mixed

TABLE I: Comparison of the three most fit individuals for each signal form class.

References

[1] L. N. d. Castro, *Fundamentals of Natural Computing (Chapman & Hall/Crc Computer and Information Sciences)*. Chapman & Hall/CRC, 2006.

[2] A. Adamatzky and B. De Lacy Costello, "Experimental logical gates in a reaction-diffusion medium: The xor gate and beyond," *Phys. Rev. E*, vol. 66, p. 046112, Oct 2002. [Online]. Available: http://link.aps.org/doi/10.1103/PhysRevE.66.046112

[3] G. Rozenberg, T. Bck, and J. N. Kok, Eds., *Handbook of Natural Computing*. Springer, 2012.

[4] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, no. 11, pp. 1021–1024, 1994.

[5] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. Knight, R. Nagpal, G. Rauch, E.and Sussman, and R. Weiss, "Amorphous computing," Massachusetts Institute of Technology, Tech. Rep., 1999.

[6] A. Turing, "On computable numbers, with an application to the entscheidungsproblem," in *Proceedings of the London Mathematical Society 1936-37*, ser. 2, vol. s2-42, no. 1. London Mathematical Society, 1937, pp. 230–265.

[7] J. von Neumann, "First draft of a report on the edvac. edited by m. d. godfrey 1992," Moore School of Electrical Engineering University of Pennsylvania, Technical report, 1945.

[8] S. Stepney, "Programming unconventional computers: Dynamics, development, self-reference," *Entropy*, vol. 14, no. 10, pp. 1939–1952, 2012. [Online]. Available: http://www.mdpi.com/1099-4300/14/10/1939

[9] P. Cariani, "To evolve an ear: epistemological implications of gordon pask's electrochemical devices," *System Research*, vol. 10, no. 3, pp. 19–33, 1993.

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

[11] J. F. Miller, S. Harding, and G. Tufte, "Evolution-in-materio: evolving computation in materials," *Evolutionary Intelligence*, vol. 7, no. 1, pp. 49–67, 2014. [Online]. Available: http://dx.doi.org/10.1007/s12065-014-0106-6

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

[13] A. Thompson, I. Harvey, and P. Husbands, "Unconstrained evolution and hard consequences," in *Towards Evolvable Hardware*, ser. Lecture Notes in Computer Science. Springer, 1995, pp. 136–165.

[14] S. L. Harding and J. F. Miller, "Evolution in materio: Computing with liquid crystal," *Journal of Unconventional Computing*, vol. 3, no. 4, pp. 243–257, 2007.

[15] C. Klegg, J. Miller, K. Massey, and M. Petty, "Travelling salesman problem solved in materio by evolved carbon nanotube device," in *Conf. on Parallel Problem Solving from Nature (PPSN)*, ser. Lecture Notes in Computer Science, vol. in Press. Springer, 2014.

[16] M. Maktuba, J. Miller, S. L. Harding, G. Tufte, O. R. Lykkeb/o, K. Massey, and M. Petty, "Evolution-in-materio: Solving machine learning classification problems using materials," in *Conf. on Parallel Problem Solving from Nature (PPSN)*, ser. Lecture Notes in Computer Science, vol. in Press. Springer, 2014.

[17] O. Lykkeb, S. Harding, G. Tufte, and J. Miller, "Mecobo: A hardware and software platform for in materio evolution," in *Unconventional Computation and Natural Computation*, ser. Lecture Notes in Computer Science, O. H. Ibarra, L. Kari, and S. Kopecki, Eds. Springer International Publishing, 2014, pp. 267–279.

[18] A. Thompson, P. Layzell, and R. S. Zebulum, "Explorations in design space: Unconventional electronics design through artificial evolution," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 3, pp. 167–196, September 1999.

[19] 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*, vol. 8, no. 4, pp. 313–317, 2012.

[20] A. Kotsialo, M. K. Massey, F. Qaise, D. A. Zeze, C. Pearson, and M. C. Petty, "Logic gate and circuit training on randomly dispersed carbon nanotubes," *International Journal of Unconventional Computing (in Press)*.

[21] L. D. Chambers, *The Practical Handbook of Genetic Algorithms: Applications, Second Edition*, 2nd ed. Boca Raton, FL, USA: CRC Press, Inc., 2000.

6.3 Paper P3

An Investigation of Square Waves for Evolution in Carbon Nanotubes Material

Odd Rune Lykkebø, Stefano Nichele and Gunnar Tufte

European Conference on Artificial Life 2015

MIT Press 2015

An Investigation of Square Waves for Evolution in Carbon Nanotubes Material

Odd Rune Lykkebø[1], Stefano Nichele[1] and Gunnar Tufte[1]
[1]Norwegian University of Science and Technology, Trondheim, Norway
{lykkebo, nichele, gunnart}@idi.ntnu.no

Abstract

Materials suitable to perform computation make use of evolved configuration signals which specify how the material samples are to operate. The choice of which input and configuration parameters to manipulate obviously impacts the potential of the computational device that emerges. As such, a key challenge is to understand which parameters are better suited to exploit the underlying physical properties of the chosen material. In this paper we focus on the usage of square voltage waves as such manipulation parameters for carbon nanotubes/polymer nanocomposites. The choice of input parameters influence the reachable search space, which may be critical for any kind of evolved computational task. We provide common measurements such as power spectrum and phase plots, taken with the the Mecobo platform, a custom-built board for evolution-in-materio. In addition, an initial investigation is carried out, which links the frequency of square waves to comparability of the output from the material, while also showing differences in the material's physical parameters. Observing the behaviour of materials under varying inputs allows macroscopic modelling of pin-to-pin characteristics with simple RC circuits. Finally, SPICE is used to provide a rudamentary simulation of the observed properties of the material. This simulation models the per-pin behaviours, and also shows that an instance of the traveling-salesman-problem can be solved with a simple randomly generated cloud of resistors.

Introduction and Background

Evolution-in-Materio (EIM) (Miller et al., 2014), (Miller and Downing, 2002) is a bottom-up approach where the intrinsic underlying physics of materials is exploited as computational medium. 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 with the aim of producing computation. Such manipulation is done with computer controlled evolution (CCE) (Harding and Miller, 2007), (Harding et al., 2008). CCE may program the materials with different kinds of stimuli, e.g. voltages and currents, temperature, and magnetic fields.

In the NASCENCE project (Broersma et al., 2012), novel nano-scale materials are being used as a substrate in which computation is attempted. In particular carbon nanotubes (CNTs) / polymer have shown promising for the solution of Travelling Salesman (Clegg et al., 2014), logic gates (Kotsialos et al., 2014), and function optimization problems (Mohid et al., 2014). To solve problems, the material is required to hold physical richness (Miller et al., 2014) under a certain manipulation scheme. The method used to manipulate the material into doing computation has until now largely consisted of setting up input signals of different kinds, e.g. static voltages (Clegg et al., 2014), square wave voltages (Lykkebø et al., 2014), a mix of both (Lykkebø and Tufte, 2014), across gold plated connectors that are exposed to the material via a glass plate. Square waves of different frequencies demonstrated potential to achieve a computationally rich behaviour (Nichele et al., 2015). As such, they are the main subject of investigation in this paper.

The hypothesis is that this electrical current exploits properties in the material that gives rise to a potentially useful output, i.e. there is an emergent behaviour emanating from interactions between current travelling through the provably non-linear material (in terms of the current/voltage relation (Massey et al., 2011) and jumps in conductivity under certain temperatures and geometric variation (Ebbesen et al., 1996)) and that measurements of this behaviour can be used for computation.

Is the material performing this computation? It can be very hard to pinpoint what computation actually is and where it happens. One key requirement is Ashby's requisite variety (Ashby, 1956), in which the importance of how many states the system can be in is underlined. In the context of computation in materials, the number of possible input states needs at least to have correspondence in physical states the material can be in. It is worth noting the difference between observable states and unobservable states. The number of observable states can be much lower than the number of states that the material can be in. When manipulating the material with an input, e.g. square waves, the material can iterate through a number of such internal states before eventually settling on the final emergent observable state, which can be in many forms, such as varying voltage peaks or phase offsets. Intuitively, these properties lead to an output of varying complexity, and an intuitively attractive idea is that the measured complexity of the output gives an indication about the internal states of the material as well; if the measured output is far more complex than the inputs, there must necessarily be a mechanism (computation) that gives rise to this complexity.

What is the best way of exploiting such computational properties? Our hypothesis is that materials with high CNT densities may act more as a conductive layer. With lower CNT densities, conductive paths may be closer to the electric percolation threshold of the CNT network (around 1% nanotubes). Another aspect that impact on computational properties is the selected range of frequencies for input signals. Specific frequencies may better penetrate the material as a result of the exploited CNT paths and signal feedbacks. Our aim is to gather basic knowledge of different material's computational properties as to be able to create a simple material model based on RC electrical circuits.

In the experiments herein, material samples with different concentrations of SWCNTs are investigated using square waves at different frequencies. Signal outputs are measured in terms of power spectrum and phase plots. In addition, compressibility of the output signals is investigated for different input frequencies and number of input pins, i.e. number of input frequencies. Such compressibility measure is close to Kolmogorov approximations (Kolmogorov, 1965), (Nichele and Tufte, 2013) of signal complexity.

Observing the response of materials with different concentrations of SWCNTs to given inputs indicate that a simple model of materials can be based on RC circuits. Such a model is developed using SPICE. The presented results provide initial thoughts regarding computational properties of used materials, together with aspects that need to be taken into account when tackling computational problems using evolution-in-materio, i.e. stability, repeatability, and noise.

The article is laid out as follows: Section II describes setup and experimental method. Section III presents the experimental results. In Section IV the modelling using SPICE is detailed and Section V provides a discussion, conclusion and ideas for further work.

Setup and Methodology

The material used for this work, i.e. carbon nanotubes/polymer nanocomposite, is placed on a microelectrode array on a glass slide which is inserted into the Mecobo board. Each sample has 16 electrodes which are physically connected to the board and can be stimulated by electrical signals, e.g. static voltages, square waves. A sketch of the experimental setup is shown in Figure 1. See (Lykkebø et al., 2014) for more details. To investigate parts of the characteristics of the material in our system under the influence of various frequencies, we run a simple frequency analysis of each material, doing a Fourier transform and find the power spectrum and phase spectrums for 3 different material samples, and at 4 different frequencies: 1KHz, 10KHz, 50KHz, 1MHz (for space reasons and to try a broad range, relative to what the board can output). The tested materials are shown in Table 1.

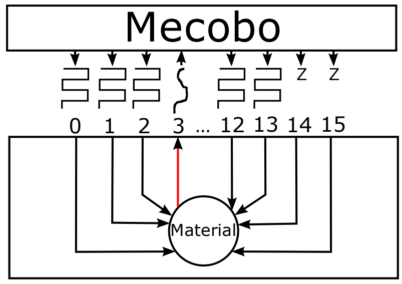


Figure 1: Experimental setup

Material	SWCNT Concentration	DC resistance
B15S01	0.75%	20k Ω
B15S02	1.00%	5k Ω
B15S05	1.75%	1k Ω

Table 1: Parameters of tested materials

All measurements were taken with the Mecobo board as described in (Lykkebø et al., 2014). Pin 3 is used as sampling pin on all measurements, and the samples were taken at 500KHz for 50 ms, thus collecting 20,000 samples.

For the compression tests (see next chapter for details), pin 3 was used as a sampling pin and input pins *to* the material were added incrementally starting at pin 0 and increasing to pin 15, skipping pin 3. We also increased the number of tested frequencies since these plots were easier to visually compress and present in this paper. The number of samples collected is deterministic. After collecting the samples, they were converted to a binary string and appended to a growing string of bits of total length 800,000 (32 bits per float), which was then passed to the built-in Python 2.7 compress()-function. The length of the compressed buffer was then measured and plotted.

Results and discussion

This Section outlines two sets of experiments. The first part deals with investigating how materials with different concentrations of SWCNTs behave when exposed to square waves at different frequencies. Output voltages, Fourier transform and phase plots are compared. In the second part, the material samples with different nanotube concentrations are analyzed again in terms of compressibility of output. As such, we may be able to relate input frequencies and output complexity in terms of compressibility for different materials.

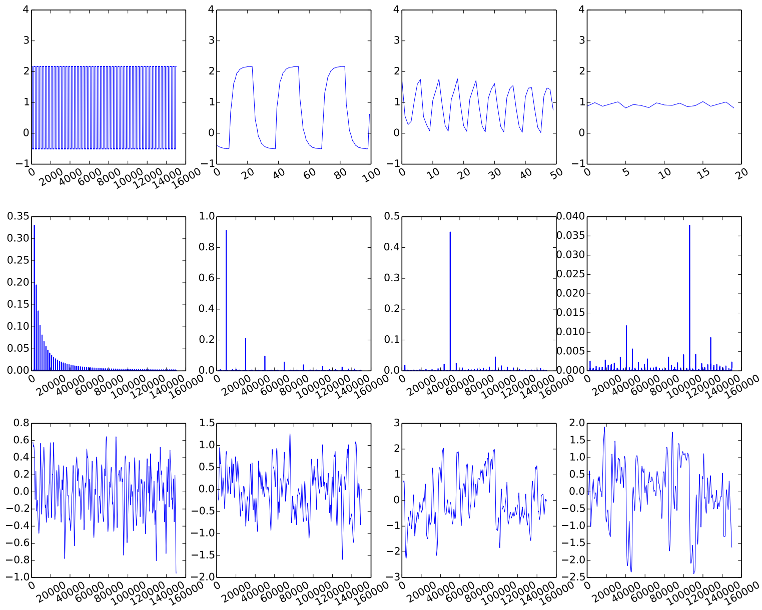


Figure 2: Material B15S01(0.75%) - Each column is a different frequency (1KHz, 10KHz, 50KHz, 1MHz). Row 1; raw voltage/time, row 2; power spectrum, row 3; phase spectrum.

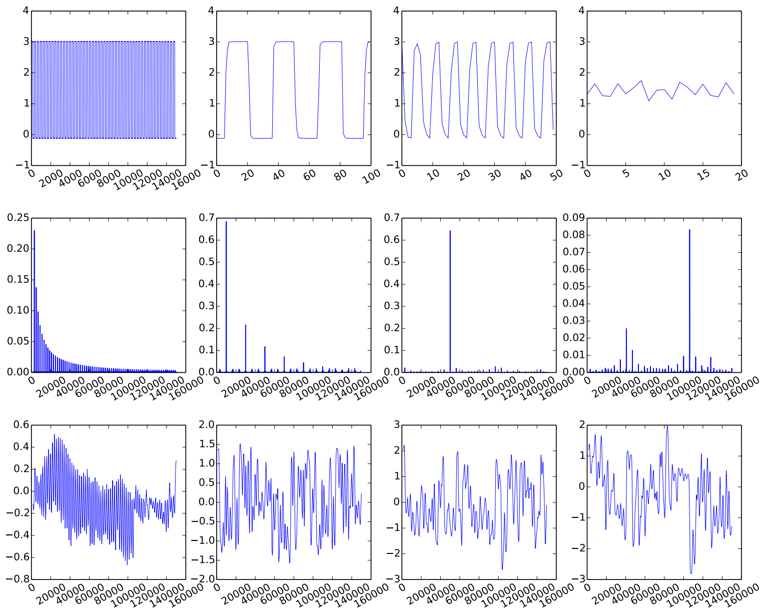


Figure 3: Material B15S02(1.00%) - Each column is a different frequency (1KHz, 10KHz, 50KHz, 1MHz). Row 1; raw voltage/time, row 2; power spectrum, row 3; phase spectrum.

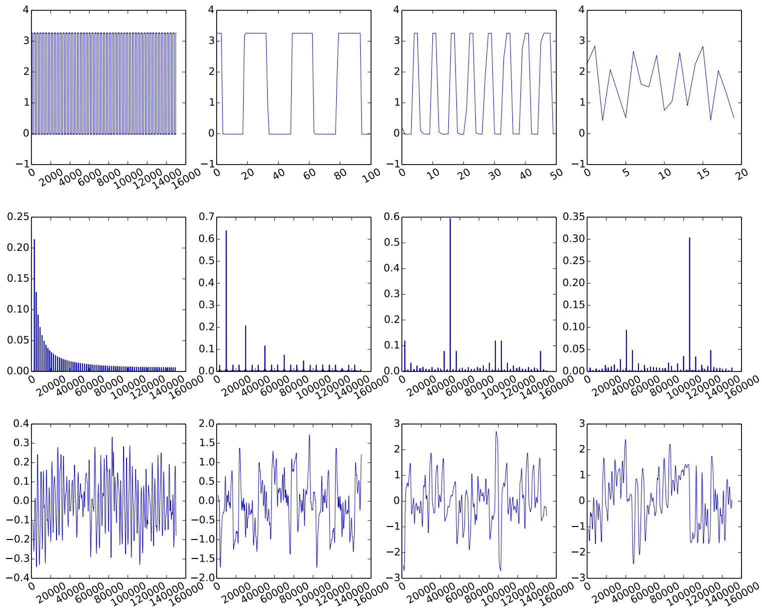


Figure 4: Material B15S05(1.75%) - Each column is a different frequency (1KHz, 10KHz, 50KHz, 1MHz). Row 1; raw voltage/time, row 2; power spectrum, row 3; phase spectrum.

Basic frequency measurements

We will first discuss Figures 2, 3 and 4. Each column corresponds to a different frequency. As can be seen from the first row in the figures, the peak-to-peak amplitude goes down as we apply higher frequencies. We believe this is the effect of capacitance in the material.

When sweeping square wave frequencies from 1Hz to 1MHz, the material initially passes the signal through (low pass), and eventually starts to let the high frequency through as well (high pass). Applying Occam's razor, the easiest way to explain this behaviour is to assume that the material has a certain amount of capacitance in addition to the easy-to-measure DC resistance. This capacitance makes the material exhibit a charge/discharge cycle when exposed to a time-varying signal such as a square wave. Another possibility is that the inductance of the material is responsible for the filtering effects. Inductance, however, is due to winding of a conductor and the distribution of the CNTs in the PMMA does not immediately appear to be laid out in a such a way as to give rise to this phenomena. Since inductive impedance increases with frequency by $j\omega L$, where L is inductance, a relatively high inductance value is required to follow the curve of the signal, as seen in Figure 12. This is also the reason why the model described below does not include any inductors; they do not seem to contribute any major factor at this observation level and at the chosen range of frequencies to investigate.

Moving on to row 2 in Figures 2, 3, 4, which shows the power spectra for each material reveals a couple of things. First, it should be noted that we have removed the DC Fourier frequency bin from the plots. Secondly, the plots show that the platform is relatively free of obvious noise. The biggest frequency responses come from the applied signals, as expected. Notice by studying the y-axis of the plots that the power goes down significantly as we increase the frequency, as is expected since the peak-to-peak voltage is also rapidly decreasing. The high-frequency plots do seem to have several more frequencies present, but this is simply an artifact of the axis scaling.

The last row in Figures 2, 3 and 4, shows the phase information from the Fourier transform. Although harder to interpret intuitively, the relatively noisy response shows that there are no obvious trends in the phase of the signals across the frequency range, which means that it is little use in trying to infer any computational properties that make use of the phase information.

Compression tests

The results from the compression test can be seen in Figures 5, 6 and 7. Note that the standard error is quite low. The main observation from these graphs is that as we add more pins that output current *into* the material, the length of the

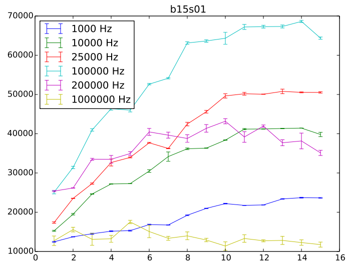


Figure 5: Material B15S01, length of compressed buffer vs. increasing number of input pins enabled.

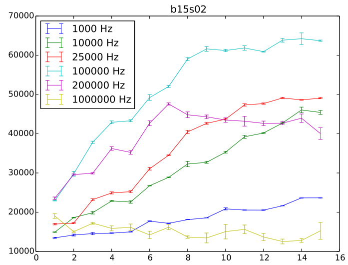


Figure 6: Material B15S02, length of compressed buffer vs. increasing number of input pins enabled.

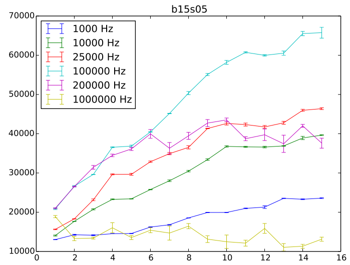


Figure 7: Material B15S05, length of compressed buffer vs. increasing number of input pins enabled.

compressed string rises sharply up to around 8-10 pins for all the materials. A threshold of saturation is further seen after this point. A possible explanation for the shape is that we add more pins, there are phase-offsets between input signals introduced due to imperfect scheduling on the Mecobo platform and potentially small interactions between the signals that give harder-to-compress output. As the number of pins increases, the amount of energy flowing through the system gives a more or less uniformly noisy measured signal. Manual inspection of the measured signal from the material confirms this.

Note that the 100KHz signal is higher than the rest. This is a combination of the measurement rate and bandwidth of the material. There is still relatively low damping seen on the signal at this rate, and 5 samples per cycle are sufficient to capture the main shape of the measured signal, and at the same time there is variation due to the frequency of output. At this frequency there is also ringing due to Gibbs phenomenon visible at high sample rates taken with an oscilloscope, giving rise to even more unpredictability.

The 1000Hz signal is almost undamped in the material, and it is therefore sharp and with a low amount of noise, making it easy to predict. The 1MHz signal is the exact opposite; very high damping ensures that the measured signal is mostly noise which should be relatively constant throughout the measurements, in particular at the sample rate we are limited to.

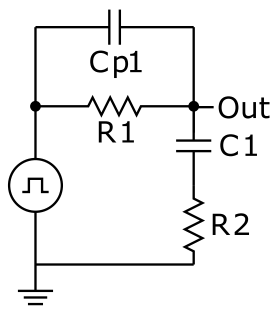


Figure 8: Simulation circuit, the output is measured from GND to the node between R1 and C1.

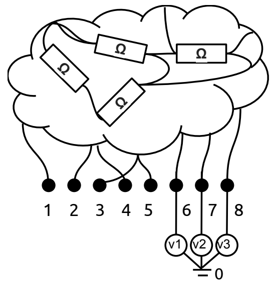


Figure 9: SPICE model used for the TSP experiments. Numbers correspond to SPICE nodes, v1,v2 and v3 are voltage sources. The cloud consists of randomly connected resistors of various values.

SPICE modelling

SPICE is a circuit simulation tool. By observing the behaviour of the material under certain inputs, it is possible to create a macroscopic 'pin-to-pin' model of a material slide. For each pin pair, we measure the DC resistance and voltage response under various frequencies, and observe the response via an oscilloscope or via the Mecobo platform. This enables us to produce models based on common circuit elements such as capacitors, inductors and resistors that mimic what we observe at this level. A practical use for SPICE models is often to tune circuits in such a way that one obtains so-called *impedance matching*, in which the impedance seen by the observing element matches its own internal impedance. This allows for maximum power in the transferred signal. In the context of evolution-in-materials, the model can be used for such a purpose as well; since the materials provide a broad range of 'loads', we can use a model of the macroscopic properties to tune our measurement circuitry in such a way as to increase our chances of

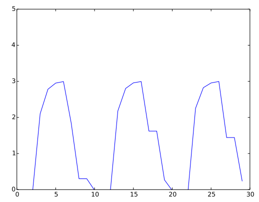


Figure 10: Mecobo capture of voltages for a 50KHz signal at 500KHz sample rate.

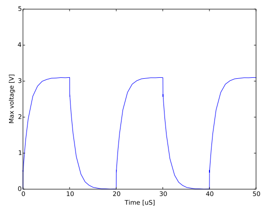


Figure 11: Spice output voltages for a 50KHz signal, simulated at 10ns timesteps.

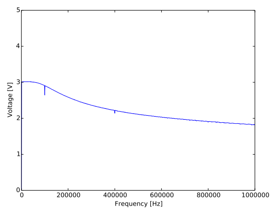


Figure 12: Mecobo capture of maximum voltage per frequency.

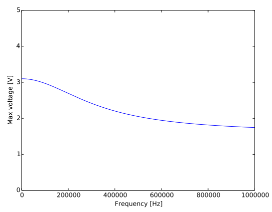


Figure 13: Spice output of maximum voltage per frequency.

finding emergent behaviours resulting from microscopic interactions in the material.

The model required to capture the effects of square waves seen at the level we are observing on can be simple. As discussed in the previous section, we believe that the main contributor to the filtering effects can be modelled by a single small filter, and this is reflected in Figure 8. By doing these measurements on all pin pairs and inserting one of these models between each pin pair, a model of the complete material slide can be constructed.

The model for this can be seen in Figure 8. This circuit is an RC low pass filter (R1 and C1) connected together with an RC high-pass filter (R2 and C2), along with a parasitic capacitance in parallel with the 'main' resistor (Cp1). The DC load of the system is captured by R1. The parasitic capacitance is added to model ringing, either due to Gibbs phenomenon (Hewitt and Hewitt, 1979) or simply parasitic capacitances and inductances resonating at their characteristic frequency). We observe these effects when applying certain frequencies to the material, which also can be barely seen in Figure 12 for the captured signal and Figure 10 for the model, albeit not to a large extent. In both cases a signal of 50KHz has been applied to the material. The signal output from the circuit is measured over C1 and R2, and the input is modelled by a pulse train voltage source. Looking at Figures 10 and 11, it is obvious that the signal captured with Mecobo (which used a sample rate of 500KHz), does not reproduce all the effects seen at simulation level, such as the ringing present due to Gibbs phenomenon and possibly other small parasitic inductances and voltages, both in the material and the measurement apparatus. We therefore used an oscilloscope to capture a more detailed version of the signal, in which more of the effects mentioned above can be observed. This begs a deeper question: are there events in the material occurring that we fail to capture with our measurements? Take inductance, as mentioned above for instance. Since there is current flowing through the material (the conductor), there must also be a magnetic field associated with this moving current. Isolated it is likely very small, but the sum of the fields could be measurable with Hall effect detectors (Ramsden, 2006), which could lead us to further improve the model by adding the correct inductances as to mirror the effects seen.

Travelling salesman

A further experiment was set up to attempt to solve the travelling salesman solved by Clegg in (Clegg et al., 2014) using a relatively simple SPICE model. In essence the problem as presented condenses into finding a way to configure the material into settling on a number voltages of different values. Each output pin is tagged with a city and the goal is to have the sorted values of the voltages on the output pins correspond to the shortest path between these cities, e.g. if pins 1, 2, 3 corresponds to the cities, and the shortest path between them is 2, 3, 1, the requirement is that $V(pin2) < V(pin3) < V(pin1)$. For details regarding the problem definition, we defer the reader to (Clegg et al., 2014).

Figure 9 shows our SPICE-model of this problem. We generate a 'cloud' consisting of nothing but resistors and connections (or 'nodes' in SPICE-terms) between them. An initial study of the problem revealed that we did not need to include capcitors to solve the problem. This is not surprising since a capacitor is an open switch with DC current. The first k SPICE-nodes (1 to 5 in the figure) are used as 'cities' and are only used as voltage measurement points. The remaining pins are connected to voltage sources whose voltages are adjusted with a (1+4) evolutionary strategy. The resistor-cloud-generating procedure uses an adjustable number of internal nodes and an adjustable number of generated resistors, giving us the ability to adjust the density of the cloud with these two parameters. In addition, the values of the resistors are drawn from a uniform distribution in a range also passed to the procedure, which enables tuning of the 'point-to-point' resistance of the network.

The instance we are seeking to provide a solution for is one of the presented solutions in (Clegg et al., 2014); a circular arrangement of 8 cities in which the shortest path is simply that– a circle. The cities are arranged on a grid, and the path length of the suggested solutions (as described above) is measured as a real number and taken as the fitness.

The vector being optimized is a 4-tuple consisting of 4 floating point values, (V_1, V_2, V_3, V_4), which maps directly to the voltages set on the voltage sources. Mutation is done by adding a number drawn from a standard normal distribution with $\mu = 0.0, \sigma = 2.0$ to one of the elements of the tuple. The SPICE simulation is run for 20ms using transient analysis with a resolution of 0.01ms. The final value of the measurements is the arithmetic average of the time series as measured across the nodes labeled as cities (i.e. nodes 1 to 8) and node 0 (which corresponds to ground).

We find that solutions are relatively quickly obtained (within 100 generations) in some of the randomly generated networks, while in others the search settles into a local optima and is unable to escape before the termination of the search which happens at 2000 generations.

This led us to investigating a few different parameter settings of the resistor cloud, the results of which can be seen in table 2. We generated 1000 networks for each parameter setting and counted the number of perfect solutions within 100 generations. As can be seen, there are relatively few solutions in all cases. Since we spent no significant time evaluating the different combinations, this is to be expected– the only conclusion that can be drawn from these numbers is that the composition of the resistor cloud matters when solving this type of problem. The somewhat unsurprising conclusion is that the composition of the materials used in the 'real' evolution in materials matters.

Nodes	Resistors	Solutions
35	150	30
35	300	19
35	500	32
60	150	69
60	300	11
60	500	34

Table 2: Results from running a search for a TSP solution in 1000 randomly generated resistor cloud networks.

Discussion and conclusions

How should we view the results in (Lykkebø et al., 2014), (Mohid et al., 2014) in which square waves are used as one of the configuration parameters then? The only observable property of the material pinpointed thus far is the fact that it will act as low pass/high pass signal filter. The different pin pairs thus act like 'frequency selectors' and it is possible to think of ways to do a form of computation with these (a frequency discriminator (Thompson, 1997) for instance), which could provide a mechanism that an evolutionary algorithm could potentially find and make use of. Another explanation that needs to be investigated further is the very real possibility that the fitness functions have been defined in such a way that they can use *noise* to solve the problem, and that lack of re-evaluations in the experiments lead to solutions that worked once, but only by chance; by encoding the solution to the problem in the input and simply evolving a way to reconstruct the solution from the input and the noise added by the material.

As for configuration of the material using static DC voltages, we have shown that the material in this case behaves mostly as a resistor network. As we have shown in our simple SPICE model shown in Section IV it is indeed possible to solve the problem instance solved by Clegg in (Clegg et al., 2014) using a network of resistors. One important point to note is that the material is rich and more flexible than one single purpose-built resistor network; it contains a large amount of various such networks, each of which can be exploited by evolution to solve a number of problems.

An issue regarding this way of formulating the problem is that we are limited by the resolution of the measurements. A common ADC such as the one used in the Mecobo daughterboard, has a resolution of 12 bits, giving a maximum of 2^{12} different voltages, effectively limiting the number of cities to 4096. This is the best case, often times one can only hope to achieve 10-11 bits of resolution, depending on how well-matched the impedance of the load is to the measurement apparatus, which again is affected by the characteristics of the material which can vary by large amounts, as we have seen previously in this paper. One can of course invest in even higher-resolution measurement apparatus to achieve well over 20 bit resolution but the point of diminishing returns in terms of practicality of building a computing system seems likely to be reached quite fast by making further advancements in this.

For the results in (Kotsialos et al., 2014) we suggest that it might be hard to reproduce these results as well. Making a XOR gate out of a resistor network is not possible since a way of inverting a signal is needed; but it is not unlikely that it is possible to achieve with a noisy system, as there are an abundant amount of transistors, diodes and let's not forget, *environmental* noise readily available in a physical system. This of course leads us into the question of where to draw the line between the computational entity, the measurement apparatus and the input. For more discussion around this, see (Lykkebø et al., 2014).

As such, we conclude that future work using the SWCNTs thus needs to take particular care with proving that it is 1) better than a random search and 2) that the results obtained are repeatable with a high degree of confidence. One way of achieving this would be to operate *within the bandwidth* of the material, such that we can be sure we are measuring actual signal response and not noise. The second, obvious way, is to always discard unstable solutions if they fail to reproduce their behaviour a number of times. A third point to note is that we must construct the fitness functions in such a way as to minimize the evolutionary process' natural tendency to exploit unwanted effects such as noise.

Much work remains before we can draw final conclusions regarding the computational properties of random carbon nanotubes. One dimension that is immediately interesting in terms of richness is movability of the material, since geometric properties play an important role in solid state physics. Introducing a more viscous environment for the CNTs to move around in could prove fruitful, since there is evidence that the tubes are capable of self-organizing (Belkin et al., 2015), and that geometry matters (Ebbesen et al., 1996) and just as a human designer is free to exploit the spatial properties of electro-material interactions, so is it possible for an evolutionary process to do the same.

Acknowledgements

The research leading to these results has received funding from the [European Community's] Seventh Framework Programme ([FP7/2007-2013] [FP7/2007-2011]) under grant agreement no [317662].

References

Ashby, W. R. (1956). *An introduction to cybernetics*. Chapman & Hall, London.

Belkin, A., Hubler, A., and Bezryadin, A. (2015). Self-assembled wiggling nano-structures and the principle of maximum entropy production. *Sci. Rep.*, 5.

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

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

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

Ebbesen, T. W., Lezec, H. J., Hiura, H., Bennett, J. W., Ghaemi, H. F., and Thio, T. (1996). Electrical conductivity of individual carbon nanotubes. *Nature*, 382(6586):54–56.

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

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

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

Hewitt, E. and Hewitt, R. (1979). The gibbs-wilbraham phenomenon: An episode in fourier analysis. *Archive for History of Exact Sciences*, 21(2):129–160.

Kolmogorov, A. N. (1965). Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1:1–7.

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

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

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

Massey, M. K., Pearson, C., Zeze, D. A., Mendis, B. G., and Petty, M. C. (2011). The electrical and optical properties of oriented langmuir-blodgett films of single-walled carbon nanotubes. *Carbon*, 49:2424.

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

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

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

Nichele, S., Laketic, D., Lykkebø, O. R., and Tufte, G. (2015). 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.

Nichele, S. and Tufte, G. (2013). Measuring phenotypic structural complexity of artificial cellular organisms - approximation of kolmogorov complexity with lempel-ziv compression. In *Innovations in Bio-inspired Computing and Applications - Proceedings of IBICA 2013, August 22 -24, 2013 - Ostrava, Czech Republic*.

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

Ramsden, E. (2006). *Hall-Effect Sensors - Theory and Application (2nd Edition)*. Elsevier.

Thompson, A. (1997). 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.

6.4 Paper P4

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

Stefano Nichele, Odd Rune Lykkebø, and Gunnar Tufte

IEEE Symposium Series on Computational Intelligence

IEEE 2015

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 nanoscale 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

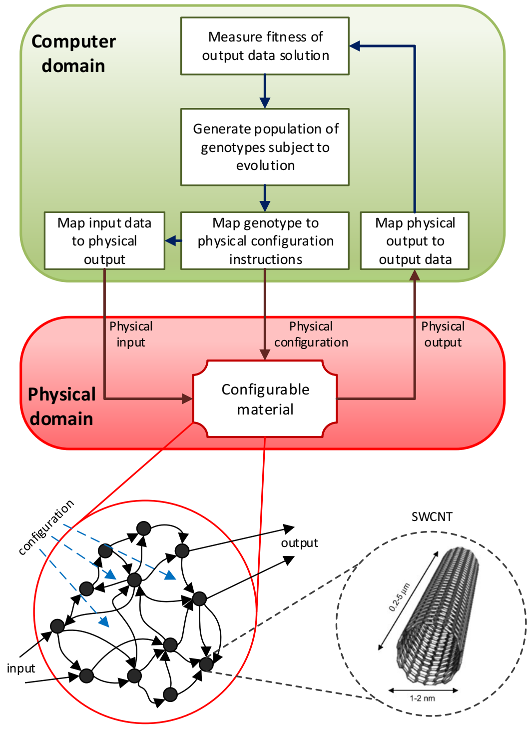


Fig. 1: Overview of Evolution-in-Materio: in the computer domain an evolutionary algorith generates configuration instruction to program the material. Those are tested against given inputs and physical outputs are mapped back and evaluated for fitness. Adapted from [4].

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.

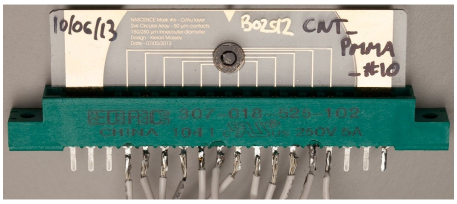


Fig. 2: Example of glass slide with microelectrode array covered by carbon nanotube / polymer nanocomposite

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.

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

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.

Input Pin1	Freq.	Duty cycle	Input Pin2	Freq.	Duty cycle	... # Inputs	Output Pin1	Output Pin2	Free Pin1	... # Free Pins
	1			2		...		...		16

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

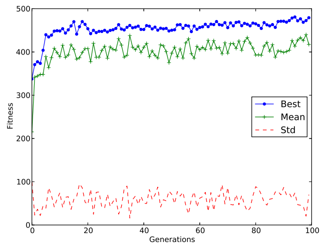


Fig. 4: 5,00% nanotubes, evolve maximum difference on 2 output pins, 1 single input.

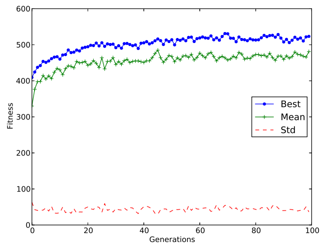


Fig. 5: 5,00% nanotubes, evolve maximum difference on 2 output pins, 5 input pins.

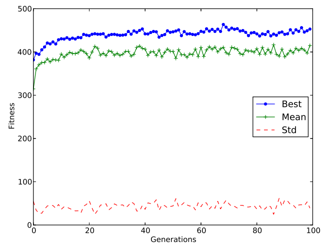


Fig. 6: 5,00% nanotubes, evolve maximum difference on 2 output pins, 10 input pins.

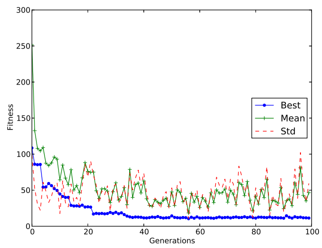


Fig. 7: 5,00% nanotubes, evolve minimum difference on 2 output pins, 1 single input.

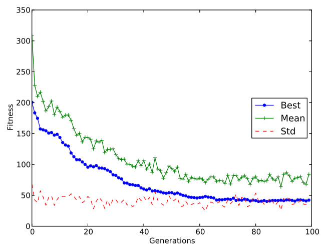


Fig. 8: 5,00% nanotubes, evolve minimum difference on 2 output pins, 5 input pins.

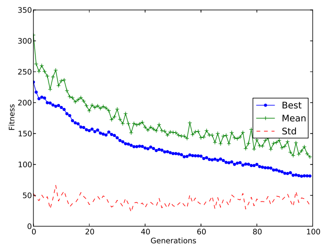


Fig. 9: 5,00% nanotubes, evolve minimum difference on 2 output pins, 10 input pins.

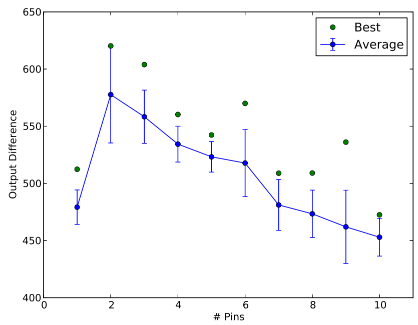


Fig. 10: 5,00% nanotubes, evolve maximum difference on 2 output pins, summary.

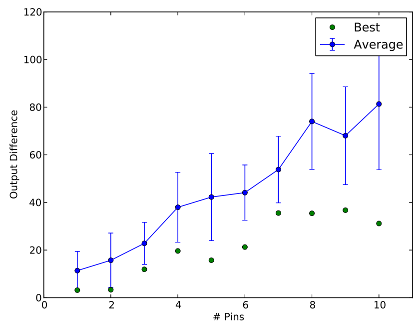


Fig. 11: 5,00% nanotubes, evolve minimum difference on 2 output pins, summary.

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

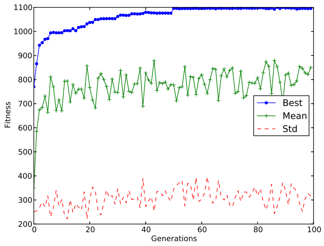


Fig. 12: 0,53% nanotubes, evolve maximum difference on 2 output pins, 1 single input.

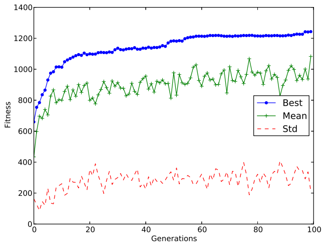


Fig. 13: 0,53% nanotubes, evolve maximum difference on 2 output pins, 5 input pins.

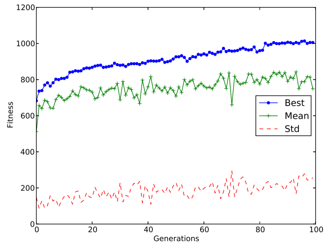


Fig. 14: 0,53% nanotubes, evolve maximum difference on 2 output pins, 10 input pins.

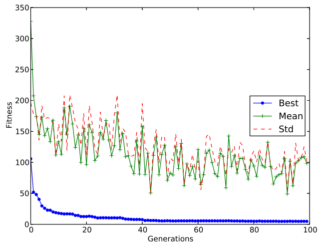


Fig. 15: 0,53% nanotubes, evolve minimum difference on 2 output pins, 1 single input.

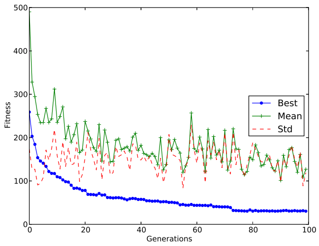


Fig. 16: 0,53% nanotubes, evolve minimum difference on 2 output pins, 5 input pins.

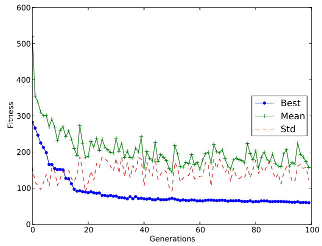


Fig. 17: 0,53% nanotubes, evolve minimum difference on 2 output pins, 10 input pins.

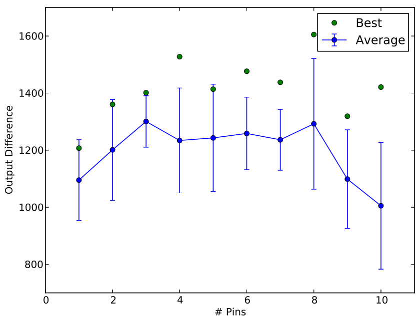


Fig. 18: 0,53% nanotubes, evolve maximum difference on 2 output pins, summary.

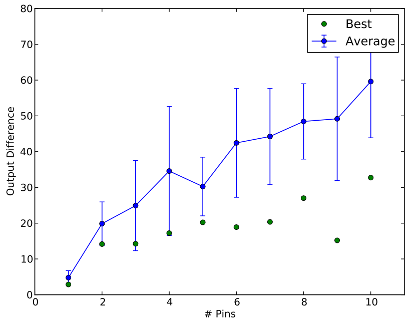


Fig. 19: 0,53% nanotubes, evolve minimum difference on 2 output pins, summary.

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

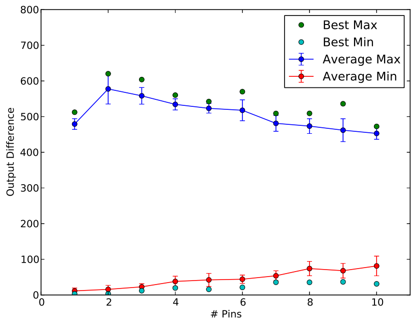


Fig. 20: 5,00% nanotubes, comparison summary, evolvability range.

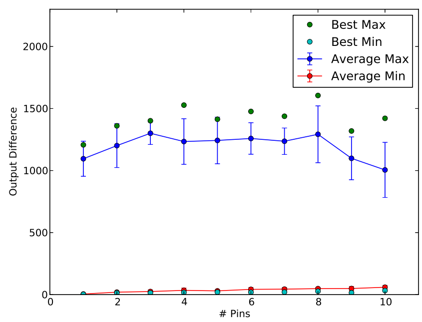


Fig. 21: 0,53% nanotubes, comparison summary, evolvability range.

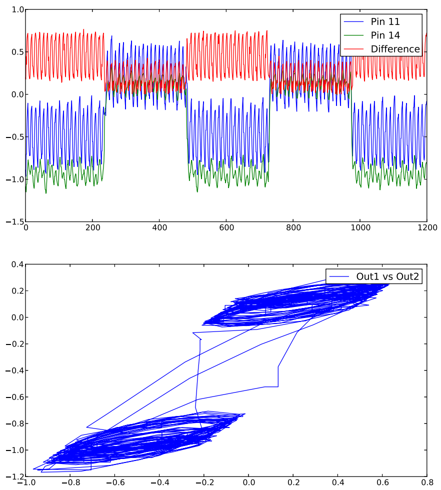


Fig. 22: Material B09S12, 0.53% nanotubes, example with 4 input pins.

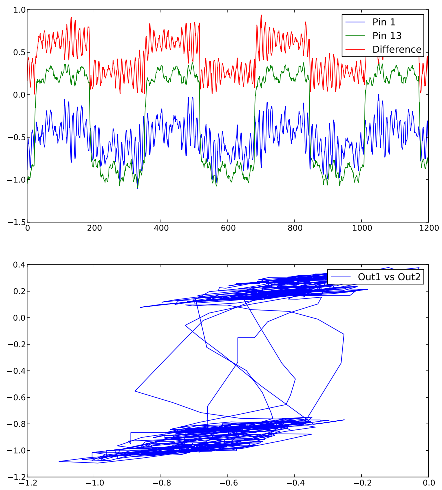


Fig. 23: Material B09S12, 0.53% nanotubes, example with 5 input pins.

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

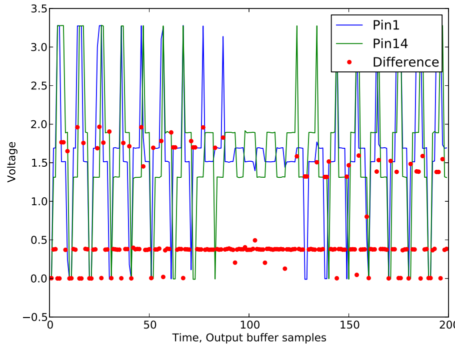


Fig. 24: 5,00% nanotubes, example with 2 inputs, evolve max difference. Input pin 13: 23945 Hz, duty cycle 27%; Input pin 2: 24576 Hz, Duty cycle 96%, Output pins: 1, 4.

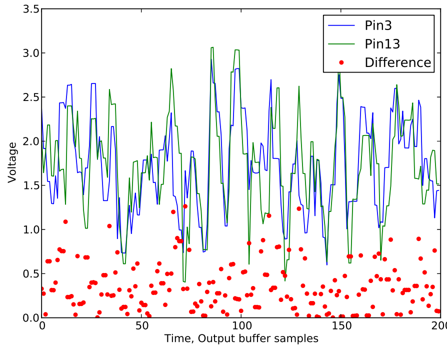


Fig. 25: 5,00% nanotubes, example with 10 inputs, evolve max difference. Input pins (pin#, frequency Hz, duty cycle %): (4, 3205, 49), (10, 23950, 3), (6, 5847, 85), (14, 24761, 46), (15, 8258, 54), (1, 6098, 83), (12, 15177, 40), (2, 19886, 3), (11, 16632, 9), (8, 22859, 99). Output pins: 3, 13.

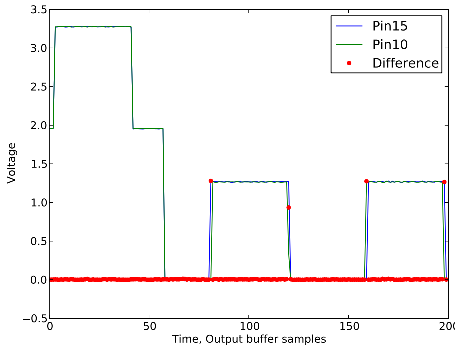


Fig. 26: 5,00% nanotubes, example with 2 inputs, evolve min difference. Input pin 4: 3208 Hz, duty cycle 94%; Input pin 13: 400 Hz, Duty cycle 79%, Output pins: 15, 10.

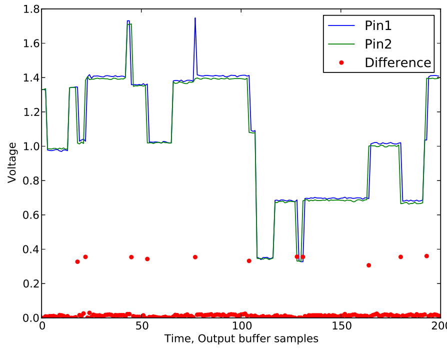


Fig. 27: 5,00% nanotubes, example with 10 inputs, evolve min difference. Input pins (pin#, frequency Hz, duty cycle %): (13, 1976, 78), (9, 3996, 11), (10, 1984, 10), (7, 657, 71), (11, 866, 13), (15, 663, 93), (12, 2060, 41), (6, 24500, 100), (14, 1462, 94), (8, 12910, 100). Output pins: 1, 2.

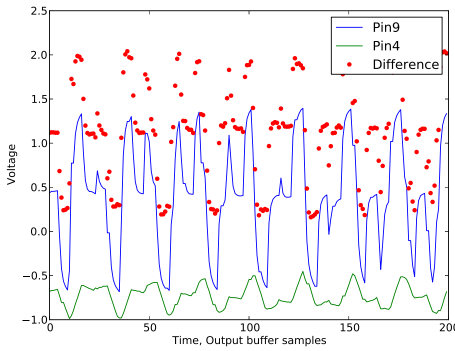


Fig. 28: 0,53% nanotubes, example with 2 inputs, evolve max difference. Input pin 10: 9595 Hz, duty cycle 79%; Input pin 8: 20299 Hz, Duty cycle 91%, Output pins: 9, 4.

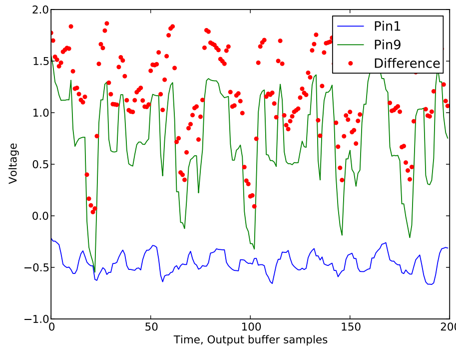


Fig. 29: 0,53% nanotubes, example with 10 inputs, evolve max difference. Input pins (pin#, frequency Hz, duty cycle %): (12, 14190, 13), (8, 22055, 30), (7, 15255, 57), (0, 13302, 100), (11, 15089, 69), (5, 6322, 21), (10, 9437, 72), (14, 7471, 55), (13, 15197, 17), (15, 9928, 29). Output pins: 1, 9.

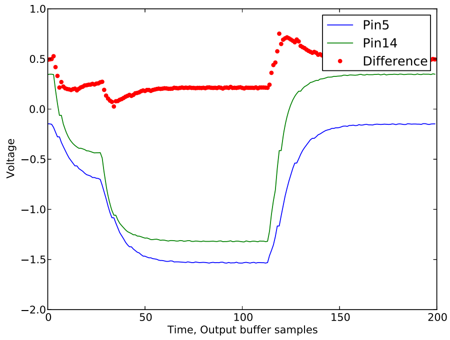


Fig. 30: 0,53% nanotubes, example with 2 inputs, evolve min difference, low frequencies. Input pin 9: 1391 Hz, duty cycle 86%; Input pin 10: 1135 Hz, Duty cycle 17%, Output pins: 5, 14.

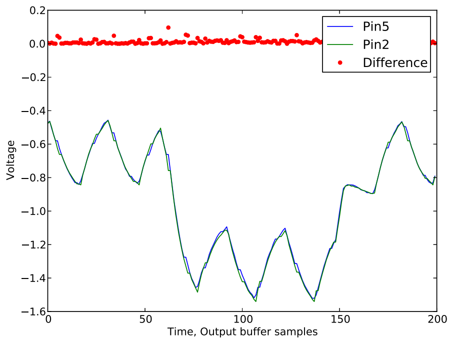


Fig. 31: 0,53% nanotubes, example with 2 inputs, evolve min difference, high frequencies. Input pin 9: 1391 Hz, duty cycle 86%; Input pin 10: 8266 Hz, Duty cycle 17%, Output pins: 5, 2.

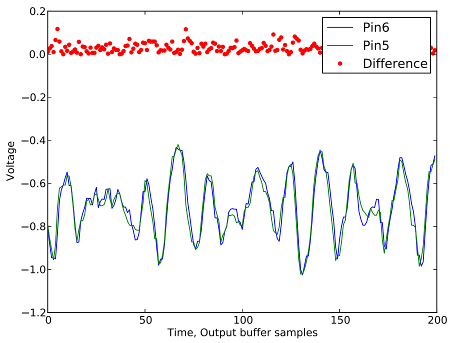


Fig. 32: 0,53% nanotubes, example with 10 inputs, evolve min difference. Input pins (pin#, frequency Hz, duty cycle %): (14, 21177, 100), (15, 23742, 44), (8, 16808, 39), (9, 480, 74), (11, 5246, 88), (10, 17164, 89), (13, 13088, 6), (0, 13358, 89), (4, 10403, 17), (12, 17164, 55). Output pins: 6, 5.

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.

6.5 Paper P5

Evolution-in-Materio of a dynamical system with dynamical structures

Odd Rune Lykkebø, and Gunnar Tufte

International Conference on the Synthesis and Simulation of Living Systems 2016

MIT Press 2016

Evolution-in-Materio of a dynamical system with dynamical structures

Odd Rune Lykkebø[1], Gunnar Tufte[1]

[1]Norwegian University of Science and Technology
lykkebo@idi.ntnu.no, gunnart@idi.ntnu.no

Abstract

Evolution-in-Materio aims to exploit real-world physics of materials to achieve computation by a combination of external stimulus and interpretation of the state of materials through measurements and observations. In a majority of Evolution-in-Materio work the dynamics of the material is filtered out, or the problem is defined in a way that the sought solution is a point attractor. In this work we explore the dynamics of materials. Within the assumption that suited materials include rich behavior emerging from the underlying physical processes there should be observable behavior similar to Dynamical Systems with Dynamical Structures $((DS)^2)$. Such behavior result in systems with a possibility of inducing perturbations to their own dynamics. Further, the importance of the *observation level* used when observing and interpreting the state of the materials is discussed and related to dynamics in Evolution-in-Materio systems.

Introduction

Evolution-In-Materio (Miller et al., 2014) (EIM) can be seen as a method to explore unconventional computation, i.e. a computer operating outside of the traditional Turing/von Neumann (Turing, 1937; von Neumann, 1993) computational model and architecture, exploiting the power of evolution, i.e. Computer Controlled Evolution (CCE) to manipulate a physical system to search for regimes where the intrinsic properties of materials provide useful computation.

The concept of EIM is, as stated by Miller et al. (2014): "to exploit the intrinsic properties of materials, or "computational mediums", to do computation, where neither the structure nor computational properties of the material needs to be known in advance (Miller and Downing, 2002). In this way evolution is a bottom-up design process that can exploit natural physical processes to do useful computation."

Herein the bottom-up design concept is further investigated so as to gain a deeper insight toward exploiting physical systems such as materials for computation: "where neither the structure nor computational properties of the material needs to be known in advance". Both the structure and the intertwined underlying physics leading to computation are products of the bottom-up design approach taken.

When a bottom-up process such as evolution acts on physical systems, there may be intrinsic processes in the computational medium that influence the computational function as a result of underlying properties resulting in a non-static structure and thereby non-static functionality, i.e a two way coupling between dynamic structure and functionality. A system with the possibility of inducing perturbations to their own dynamics as a function of their system states, enables state space trajectory changes and topological reconfigurations of the state space (Omholt, 2013).

In this paper we show that such behaviours, present in living systems, can also be found in EIM systems making EIM a physical realisations of Dynamical Systems with Dynamical Structures $((DS)^2)$ (Spicher et al., 2004). In such systems state transition functions and the set of state variables can change over time.

Reaction diffusion systems as in Adamatzky (2009) work exploit massively parallelism of state updates in growing patterns where information processing take place. The reaction-diffusion computer is based on local interactions and change of spatial properties over time. Growth of patterns change the state and topological properties of the machine providing a similarity to the $((DS)^2)$ dynamical reconfigurations of the state space.

Further, it can be useful to compare the concept of EIM with morphological engineering (ME) Doursat et al. (2013). Central to ME is the concept of *agents* and the mechanisms involved in controlling them. EIM places less focus on the interaction of the individual agents, and more focus on exploitation of the emergent behaviours of the physical systems under study. Though local interaction between smaller parts of the physical matter (i.e. electro-chemical interaction between molecules or concentrations of chemicals) is crucial, EIM attempts to take a step away from the agent-focus and consider emergent properties a primary unit of study. However, one of the central aspects of ME, i.e. "endowing physical systems with information" Doursat et al. (2013) does capture EIM. 'Invisible' dynamics of physical systems can be manipulated without direct access to the individual agents' rule sets by giving the systems the ability to filter or react to information and energy presented through physical interfaces.

In this article, an EIM system is observed in a $(DS)^2$ system view. The results show that such a view is applicable and can be used to gain further insight in the working of EIM systems. As a result EIM may share several properties with self-organizing systems. Herein artificial developmental systems (Kumar and Bentley, 2003) are used as a starting point for exploring and exploiting computational mediums where computation is a product of underlying dynamical physical processes.

An implication of "exploiting the intrinsic properties of materials" is that self-organizing processes and the resulting computation do not necessary comply with our mostly used computational paradigm of digital computers. EIM can be seen as a hybrid variant of Analog Computation (AC) (MacLennan, 2007). In AC, the fact that a mathematical function provides a model of the *observed* behaviour of a physical system is used 'conversely'; a physical system can be used to calculate a mathematical function since a physical system can be parameterized and adjusted to match a large number of mathematical functions. Configurable Analogue Processors (CAP) or Field Programmable Matter Arrays (FPMA) (Miller and Downing, 2002) are recent example of this paradigm. The hybrid approach of EIM include the computational matter, e.g. CAP or FPMA, in a mixed signal system using a digital computer to configure and communicate with the material.

This is an approach that enables the computational power of the material with the ease of programmability of digital computers (Broersma et al., 2012). In a hybrid approach observability is a core issue. Ensuring that the data from the material are observable and sound without using more computational power for the observation then the actual computation (Bremermann, 1962). A practical implication of observability is that there is a need for choosing what the smallest possible change is to be on the top-level. Top-level being the level that the observable computational result(s) emerge from the underlying physical processes.

Shannon (1941) provided an early theoretical model of analogue computers, the General Purpose Analog Computer (GPAC). Though application of the GPAC model on EIM seems un-natural since it relies on chaining together components (reminiscent of agents with particular tasks) and defining bigger expressions out of this. Neural networks is yet another potential model and link between the physical EIM system and a theoretical model– this link is further discussed in Broersma et al. (2012).

Figure 1 shows an EIM system using a digital computer to host an EA to configure a material for computation. The material operates in the analogue/physical domain and the computer responsible for input/output mapping and configuration will operate in the digital domain.

As stated, our focus is to explore EIM in a $(DS)^2$ view.

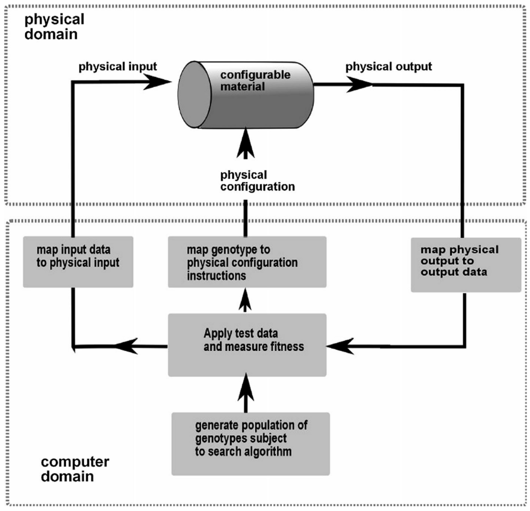


Figure 1: Principle of EIM using a hybrid approach. Taken from Miller et al. (2014).

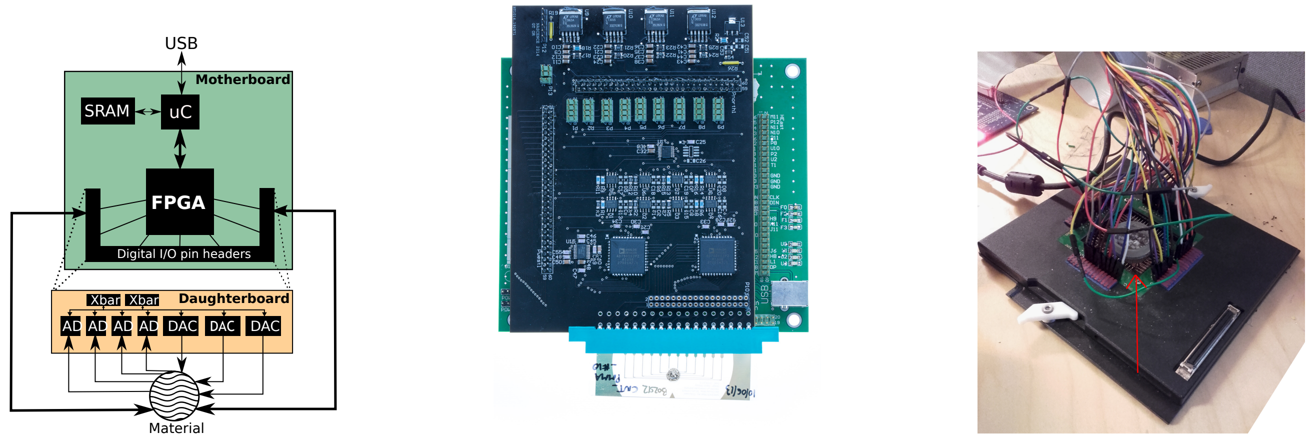
In Figure 1 the configurable material is the self-organising system. The material inhibits dynamical properties and respond to perturbations from the input and configuration signals. To observe dynamical properties, the trajectory of the system is used as a quantifiable measurement of behaviour. In artificial development similar measurements have been used as a measurement of evolved complexity (Nichele and Tufte, 2013) and to show intertwined influence of structure and computation ($(DS)^2$) on artificial developmental systems (Tufte, 2009).

The systems herein are all observed at an electrical level. The underlying physical (and electrical) processes may change over time on the microscopic level, however our observations is on the 'digital level'. As such, the goal herein is to be able to exploit the power of EIM without a high cost in ensuring the correctness of the observation, i.e. a underlying rich physical system observed in the constrained digital domain.

Background

Evolution-in-materio extended to $(DS)^2$

Gordon Pask's pioneering work in EIM(Pask, 1959) is an interesting piece of work if viewed in a $(DS)^2$ setting. Pask observed (by eye) and evaluated (by ear) the growth of neural like structures using an electrochemical device made of a dish with electrodes covered by a metal salt solution. By adjusting the current between electrodes Pask was able to grow iron connections that responded to different frequencies. Pask's system show the principles of a $(DS)^2$ EIM system. The growth of the connections depend on current, when current passes the connection grow. By adjusting the current in different regions a structure emerges and are adjusted toward the evaluator's (here Pask) goal. The



(a) Block diagram of the Mecobo hardware interface.

(b) Picture of the Mecobo motherboard with mixed signal daughter board.

(c) The material bay filled with salt crystals (red arrow).

Figure 2: Overview of the Mecobo hardware interface.

growth of the connections are governed by self-organization and perturbations. The system is constantly changing making the response to perturbations depending on the systems present state. In Pask's work this property of change by self-organization and perturbations was exploited toward the goal of growing "an artificial ear".

In the work of Clegg et al. (2014) a material of Carbon Nano Tubes (CNT) was exploited to solve the travelling salesperson problem by evolving statical configurations that manage to solve the problem. i.e. the material's dynamic properties was filtered out by allowing the system to reach a steady state, or point attractor. Thompson's experiments using a Field Programmable gate Array (FPGA) as a material (Thompson et al., 1999) evolved a static configuration defining the internal circuit architecture. The input signal (similar to Pask's) was two frequencies that changed the output value depending on the input frequency (a frequency discriminator). The static configuration in Thompson's FPGA experiments does not allow for dynamic structures. The input signal perturbed the system to an binary observable output. However, the underlying dynamics (internal state transitions) for the system was untested.

In Harding and Miller's EIM work utilizing Liquid Crystal Displays (LCDs), the material is viewed as a type of static device with an evolved configuration that enable the LCD to compute, e.g. a frequency discriminator (Harding and Miller, 2004). However, the LCD is not a static device. The behaviour change if it is disconnected and reconfigured. To regain previous behaviour a short re-evolution was required.

Systems that explicitly exploit dynamic structures such as slime moulds (Adamatzky, 2016) or the combination of CNTs and liquid crystals (Massey et al., 2015b) show a clear connection between dynamic structure and computation similar to a $(DS)^2$ system.

As in Pask's work, the work on explicit dynamic structures and the shown change in response on configuration for the LCD it can be argued that a closer look at EIM systems as a $(DS)^2$ system are fruitful as to achieve more complex behaviour in such computational medium. More over, including more knowledge of the underlying bottom-up processes ($(DS)^2$) enables an increased understanding and insight in evolutionary exploitation of EIM systems toward complex computational tasks.

Materials

Recently the NASCENCE (NAnoSCale Engineering for Novel Computation using Evolution) project (Broersma et al., 2012) has provided a variety of material samples based on CNTs (Massey, 2013), mix of liquid crystals and CNTs (Massey et al., 2015b) and gold nano particles (Boses et al., 2015) for EIM research. Studies of Single walled carbon nano tubes(SWCNT) (Massey, 2013; Massey et al., 2015a) have shown that these CNTs have novel electrochemical properties and have the potential to do computation (Massey et al., 2015b).

The results of NASCENCE (e.g. (Mohid et al., 2014; Clegg et al., 2014; Massey, 2013)) have shown that computational results can be achieved without explicitly exploiting the dynamics of physical systems. Here the exploration of $(DS)^2$ expand our target behaviour of materials to a physical system with rich dynamic properties and complexity at many scales, which is a point frequently brought up in the complex systems literature. Often one divides a system into two broad categories relative to the observation level or scale, (Sayama, 2015) (Bar-Yam, 1997) (Fromm, 2004). Microscopic properties or behaviours are those that potentially give rise to emergent macroscopic properties. At different levels of observation, different macroscopic behaviours (such as emergence and self organization) exist, and thus the complexity of a system depends on the level of observation; as Simon (1962) writes, "How complex or simple a structure is depends critically upon the way in which we describe it."

A core idea in EIM is that it is should be possible to manipulate the microscopic behaviour of the material by providing input and 'configuration' energy (which potentially affects all scales of the system) such that the emerging or self-organizing behaviour is both observable and useful in terms of computation.

To further drive the point of the richness the materials used in the experiments are chosen to be very different. Single walled carbon nano tubes in a static physical configuration, i.e. electrical charge change cause dynamics, used and exploited for computation within the NACSENCE project show electrical observable response from underlying electrical networks exploited by evolution to emerge at the macroscopic level. As a second material common kitchen table salt, a material that is conductive when mixed with water. The crystalline form of salt is for the most part non-conductive since the ions are bound up in a crystal lattice structure, but by adding a tiny amount of water to the crystals conductivity is achieved. The salt structure is in contrast to the CNT material not static.

Method

The overall goal of EIM is to find methods and materials that can serve as complex computing systems. Methods should be capable of exploring and exploiting materials toward achieving useful computation, and materials should have inherent properties that enable this. The systems are physical systems, hence the material will operate in a real environment, and in particular, the environment will be part of the system. The main method is the bottom-up design approach of evolution. In a complexity setting evolution is argued to be a process that builds complexity (Holland, 2012). The building of complexity can also be considered as a case where the system learns to program itself, third of Brian Arthur's methods of complexity growth (Arthur, 1993). Further, the concept of growth of complexity fits well into a $(DS)^2$ setting; a system that evolve toward more complex dynamic behaviour by perturbations from the environment and the dynamics of the system itself (Nichele and Tufte, 2013).

An experimental approach is taken to investigate and explore the relation between EIM and $(DS)^2$. Exploiting evolution to configure materials with a behaviour that show induced perturbations to it's own dynamics. The experimental setting is in principle as shown in Figure 1. The experiments are designed to unveil the intertwined influence between the dynamics of underlying physics the input data and configu-

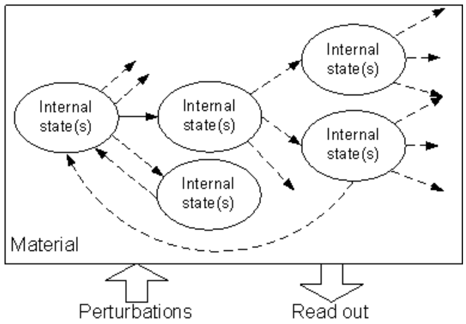


Figure 4: System view of the experiments. The material is considered as a $(DS)^2$ system capable of inducing perturbations to it's own dynamics. The external perturbations indicated as an input arrow include input data and configuration signals for the material. The output arrow indicates the external observation of the system state.

ration signals. To put the system in a $(DS)^2$ setting input and configuration signals are considered as external perturbations to the system. Figure 4 illustrates the experimental setting at a system level.

The material in the figure include a set of state transitions. The state transitions can be perturbed externally. Each state in the figure is observable through the read out signal. If the system inhibits $(DS)^2$ behaviour it should be possible to detect different trajectories (induced perturbations to it's own dynamics) whilst the external perturbation is unchanged, illustrated by dotted arrows. In the figure each state is as indicated one or several internal states (internal state(S)) as to illustrate the property of topological reconfigurations of the state space.

In literature on artificial (and biological) evolution (AE) many terms are used for breaking down the various parts of a search. In a physical and 'real' system such as the EIM-based ones it is not immediately clear where the boundaries go between these parts, such as *genotype*, *phenotype* and the *genotype-phenotype map*. The definition of these terms is context-dependant, and EIM mixes two contexts; the context of artificial evolution and the context of physical, real life systems. For the purposes of EIM it is sufficient when discussing these terms to simply note that when we discuss the *phenotype* of a system, we are talking about the observed entity interacting with the physical environment, e.g. the electrical current flowing through the material, and observed as state space trajectories, i.e. the observable read out in Figure 4. Since the material is a part of the environment by virtue of being a physical object, there is inevitable interaction between the material and the environment. The genotype can still be treated as is common in AE; the entity operated upon by genetic operators such as mutation and crossover.

The relation between the genotype and phenotype is the genotype-phenotype map, This map takes a genotype as in-

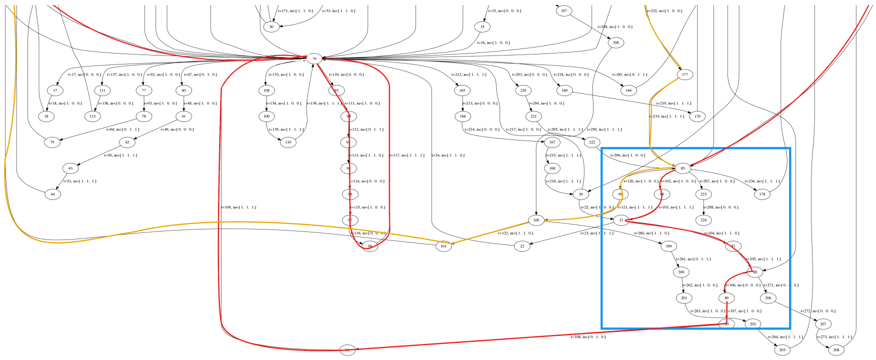


Figure 3: As an illustration of the experimental results a section of the full state space trajectory for an experimental run on salt/water solution with . A cut out in blue highlight state space trajectory changes (presented in Figure 5).

put and transforms it to a real-world entity with real-world physics and in particular includes the interface used to produce the desired manipulative phenomena, such as digital-to-analog converters. It is however worth noting that only what is observable to the EA (phenotypic behaviour or environmental effects of it's existance) can be used as input to a fitness function.

The experimental platform

The experimental results in this paper were achieved using the Mecobo platform (Lykkebo et al., 2014), a hardware/software implementation of an EIM system. The Mecobo platform is shown in Figure 2.

Figure 2(a) show the overall design. Configuration specification, i.e. genotypes, are loaded from a PC to Mecobo over the USB port. The micro controller communicates with the USB interface and with an FPGA on an internal bus. The FPGA can interface directly to materials or as in the figure use a daughter board to extend the signal range as shown in Figure2(b).

Mecobo is capable of controlling close to 100 individual configurable input/output signals (pins) that connects to the material. Each signal are described by parameters at a given point in time., e.g. recording pin from time 0, output frequency pin from time 0 to 10 or output pin voltage level 2.7V from time 0 etc., see (Lykkebo et al., 2014) for a detailed presentation of the Mecobo hardware and software.

The material samples are placed in a material bay, as seen in 2(c), pointed to with a red arrow, and connected to the Mecobo platform.

A standard genetic algorithm with tournament-based selection of size 3, a population size of 30 and a mutation probability of 0.2 was used. The mutation is drawn from a Gaussian distribution with $\sigma = 1$ and $\mu = 0$. The genome consists of 3 floating point numbers, from 0 to 1 which are scaled to integer *square wave* frequencies by 10^6 during the genotype-phenotype mapping process.

Each individual is run 5 times and stability (i.e. repeatability of the measured) states is part of the fitness. One run collects T state vectors $S(t)$ in a vector B_r, $r \in [0, 4]$.

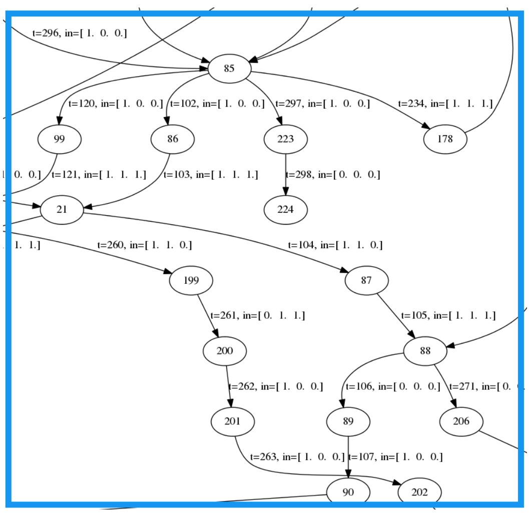


Figure 5: Zoomed region in blue in fig 3, showing the time-dependant behaviour of a driven evolved system in salt.

The fitness function counts the number of unique states measured during one time period T, where unique means that for all pairs $(t_1, t_2) \in [0, T]$ $S(t_1) \neq S(t_2)$ and divides the number of unique states by T. Finally, we take the cosine between all pairs of B_r-vectors and multiply the fitness by this number, ensuring that systems who has very similar trajectories in state space get awarded a high fitness.

The material bay has room for 60 connectors, of which we used 50 to connect to the Mecobo platform. 3 pins were then selected as designated input (*to the material*) pins; spread out in the material, one pin was selected as a current sink and the remaining 46 pins were used as material output pins, whose *digital* values were recorded over a time period of 200 uS at 20KHz.

The output pins are connected directly to the FPGA input buffers and further directly connected to FPGA-internal flip-flops with a triggering voltage of 1.7V. We define a *state* as a vector of size n $S(t) = (o_1(t), o_2(t), ..., o_n(t))$ where $o_i(t)$ is the value of flip flop i in the FPGA at time t.

We do not claim that the targeted system does useful computation. The purpose of this example is to demonstrate the existence of potentially complex behaviour in a driven physical system, which gives *potential* for useful computations. The chosen trajectory metric for the fitness evaluation is mainly chosen to be able to investigate for $(DS)^2$ behaviour. However the metric is in accordance with an abstract measurement of complexity as used by Langton (1991), Wolfram (1984) and for developmental systems (Kowaliw, 2008; Nichele et al., 2016).

Material Samples

The 'material cup', in fact a Multichannel Systems micro electrode array (MEA) model (60MEA100/10iR-Ti), pointed to in red in Figure 2(c), holds salt crystals formed by letting a solution 10mL of water with 1mL of kitchen table salt dry out, and before each evolutionary run these salt crystals are mixed with 50μL more of water to allow charge movement in the crystals. A second such MEA was filled with single walled carbon nano tubes (SWCNT) in a PMMA polymer solution. The same experiment is run on both. For control purposes, a fully conductive carbon plate was used.

Results and discussion

The Genetic Algorithm (GA) was used to provide data to be analyzed in a $(DS)^2$ setting. The resulting phenotypes from the evolutionary runs were analyzed by examining the state space traversal. In all results, $(DS)^2$ behavior was found. Figure 3 shows parts of a full state space traversal on dry salt crystals, visualized in graph form. Each node represents one state, and each arc between states one state transition. Figure 5 shows a zoomed version of the box marked by blue in figure 3. Figures 5 shows out-takes of the more interesting dynamic (time-dependent) behaviors in the run. Each edge is marked with the time-step (t) that the transition occurred on, along with the input as a tuple of 3 binary values (a,b,c). For a given input and a state one could expect that a transition should go to the same state, but as we can see for instance in state 85 of figure 5 this is not the case: at t=102, in=(1,0,0) there is a transition to state 86, whereas for t=120 the transition goes to a different state, 99. The red line in 3 traces out the full path of this branching behavior, turning orange at the second pass through state 85. This demonstrates a time-dependent behavior where the traversal of the state space depends on previously seen states.

The same method and set-up was used on the carbon nano tube(CNT) material sample. Time-dependent behavior whilst traversing the state space was present and found in all runs with the carbon nano tube material sample as well.

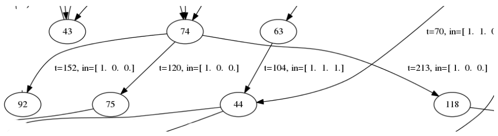


Figure 6: A state graph of a time-evolved carbon nano tubes driven state graph.

Figure 6 shows an out-take with dynamic (time-dependent) behaviors from the full state space traversal graph.

The GA typically achieves a fitness of 0.75 out of a 1.0 max within 50 generations, meaning that it finds a stable behavior that generates roughly $3/4$ unique states relative to the chosen observation level. This result is achieved in all cases tested. The number of transitions that show the time-dependent branching behavior discussed in relation to figure 6 is typically around 5. The chosen genome most often would map to frequencies around 100 KHz., which is higher than the Nyquist-rate relative to our sampling frequency of 50KHz, meaning that we cannot fully reconstruct the input signal from the sampling rate, however this simply underlines our previous points relating to where one sets the observation level– we are not concerned with signal reconstruction, but rather the systems ability to produce behavior in the $(DS)^2$ context.

This behavior is further shown in figure 7 for the salt crystals, and the plot is similar when using CNT as a material. The plot shows periodic behavior interspersed with spurious stable (in the sense that they meet our stability criteria previously defined), states. We stress that the nature of these states can have several reasons, (i.e. metastability in the flip-flops) and we cannot fully rule out the possibility of the sampling apparatus 'interfering' with the dynamics of i.e. the stimulus of the salt crystals, however runs with a fully conductive carbon plate as material gives us no such observed dynamics.

The vertical axis on this figure indicates the time as the voltage is applied to the salt crystals. On the left hand side we see the input to the system as it is captured by the same method that captures the rest of the state data– it is of a more regular nature (though some sampling artifacts).

The complexity of the input compared to the complexity of the output is currently under investigation, however for the purposes of this article the fact that the number of observable states (2^3) is much lower than the number of potential observable output states (2^{46}) demonstrates that there is potential for underlying $(DS)^2$ dynamics. Input is also verified as 'stable' in the same sense as output in that it is also based on a number of repeats of the same run and compared.

Conclusion

Analyzing the behavior of the evolved systems in a $(DS)^2$ setting show that both material samples explored show the property of inducing perturbations to their own dynamics as a function of their system states. In the state space traversal graph this behavior is visualized and show that the underlying dynamics of the materials enables changes in the trajectory in the state space. For the CNT sample the change in trajectory is a product of electrical effects that emerges as observable $(DS)^2$ traversal of the state space. For the salt crystal/water sample topological reconfigurations are possible but not detectable directly. The chosen observation level is here also only based on sampled voltages, a result of currents and charge in the material.

In the experiment *dynamics* was explicitly targeted, in contrast to previous EIM work on similar CNT materials ((Clegg et al., 2014; Mohid et al., 2014)), as the results show the hybrid approach with a digital read out dynamic behavior are present. The presence and possibility to observe such behavior can expand the computational range of a relative simple EIM-set-up from problems requiring only feed forward networks, e.g. (Clegg et al., 2014), to computational tasks requiring memory.

The $(DS)^2$ behavior at our chosen observation level show that it is not given that the system will behave predictably in the sense that a given input and given state will produce the same output– the current state might look the same, but actually be a result of a different underlying dynamic process. This does not imply that it is useless to use this system as a basis for building computational systems or similar, but rather that care must be taken when choosing and applying stimulus to the system and also when we evaluate the output by using a fitness function in a artificial evolution-approach. We suggest allowing the computational system to 'unfold' over time, treating the apparent weakness of unpredictability more as a strength. A way of doing this would be to not consider the system using reductionism and divide it's functionality into smaller parts (i.e. individual gates), but rather consider the system as a whole as a 'basic component' and it's dynamic properties a way of achieving computation.

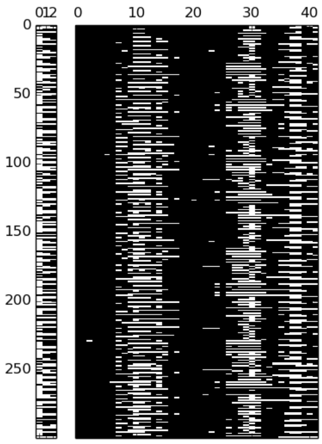


Figure 7: Salt crystals state plot

References

Adamatzky, A. (2009). *Encyclopedia of Complexity and Systems Science*, chapter Reaction-Diffusion Computing, pages 7548–7565. Springer New York, New York, NY.

Adamatzky, A. (2016). *Advances in Physarum Machines*. Springer International Publishing.

Arthur, W. B. (1993). On the Evolution of Complexity. Working Papers 93-11-070, Santa Fe Institute.

Bar-Yam, Y. (1997). *Dynamics of Complex Systems*. Studies in Nonlinearity. Westview Press.

Boses, S. K., Lawrence, C. P., Liu, Z., Makarenko, S., van Damme, R. M. J., Broersma, H. J., and van der Wiel, W. G. (2015). Evolution of a designless nanoparticle network into reconfigurable Boolean logic. *Nature Nanotechnology*, 10(12):1048–1052.

Bremermann, H. J. (1962). *Self-Organizing Systems-1962*, chapter Optimization through Evolution and Recombination, pages 93–106. Spartan Books.

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

Clegg, K., Miller, J. F., Massey, M. K., and Petty, M. (2014). Travelling Salesman Problem solved in materio by evolved carbon nanotube device. In *Parallel Problem Solving from Nature – PPSN XIII: 13th International Conference*, Lecture Notes in Computer Science, pages 692–701. Springer.

Doursat, R., Sayama, H., and Michel, O. (2013). A review of morphogenetic engineering. *Natural Computing*, 12(4):517–535.

Fromm, J. (2004). *The Emergence of Complexity*. Kassel University Press.

Harding, S. L. and Miller, J. F. (2004). A Tone Discriminator in Liquid Crystal. In *Congress on Evolutionary Computation(CEC2004)*, pages 1800–1807. IEEE.

Holland, J. H. (2012). *Signals and Boundaries: Building Blocks for Complex Adaptive Systems*. MIT Press.

Kowaliw, T. (2008). Measures of complexity for artificial embryogeny. In *GECCO '08: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 843–850. ACM.

Kumar, S. and Bentley, P. J., editors (2003). *On Growth, Form and Computers*. Elsevier Limited Oxford UK.

Langton, C. G. (1991). Computation at the edge of chaos: phase transitions and emergant computation. In Forrest, S., editor, *Emergent Computation*, pages 12–37. MIT Press.

Lykkebo, O. R., Harding, S., Tufte, G., and Miller, J. F. (2014). Mecobo: A hardware and software platform for in materio evolution. pages 267–279.

MacLennan, B. J. (2007). A Review of Analog Computing. Technical Report UT-CS-07-601. Technical report, University of Tennessee, Knoxville.

Massey, M., Volpati, D., Qaiser, F., Kotsialos, A., Pearson, C., Zeze, D., and Petty, M. (2015a). Alignment of liquid crystal/carbon nanotube dispersions for application in unconventional computing. *AIP Conference Proceedings*, 1648(1).

Massey, M. K. (2013). *Electrical Properties of Single-Walled Carbon Nanotube Networks Produced by Langmuir-Blodgett Deposition*. PhD thesis, Durham University, UK.

Massey, M. K., Kotsialos, A., Qaiser, F., Zeze, D. A., Pearson, C., Volpati, D., Bowen, L., and Petty, M. C. (2015b). Computing with carbon nanotubes: Optimization of threshold logic gates using disordered nanotube/polymer composites. *Journal of Applied Physics*, 117(13):134903.

Miller, J. F. and Downing, K. (2002). Evolution in materio: Looking Beyond the Silicon Box. In *2002 NASA/DOD Conference on Evolvable Hardware*, pages 167–176. IEEE Computer Society Press.

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

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

Nichele, S., Giskeødegard, A., and Tufte, G. (2016). Evolutionary Growth of Genome Representations on Artificial Cellular Organisms with Indirect Encodings. *Artificial Life*, 22(1):76–111.

Nichele, S. and Tufte, G. (2013). Evolution of Incremental Complex Behavior on Cellular Machines. In *ECAL 2013*, pages 63–70. MIT Press.

Omholt, S. W. (2013). From sequence to consequence and back. *Progress in Biophysics and Molecular Biology*, 111(2-3):75–82.

Pask, G. (1959). Physical analougues to the growth of a concept. In *Mechanisation of Thought Processes*, number 10 in National Physical Laboratory Symposium, pages 877–922. Her Majesty's Stationery Office, London, UK.

Sayama, H. (2015). *Introduction to the Modeling and Analysis of Complex Systems*. Open SUNY Textbooks.

Shannon, C. E. (1941). Mathematical Theory of the Differential Analyzer. *Journal of Mathematics and Physics*, 20:337–354.

Simon, H. a. (1962). The architecture of complexity. *American Philosophical Society*, 106(6):467–482.

Spicher, A., Michel, O., and Giavitto, J.-L. (2004). *Cellular Automata: 6th International Conference on Cellular Automata for Research and Industry, ACRI 2004, Amsterdam, The Netherlands, October 25-28, 2004. Proceedings*, chapter A Topological Framework for the Specification and the Simulation of Discrete Dynamical Systems, pages 238–247. Springer Berlin Heidelberg, Berlin, Heidelberg.

Thompson, A., Layzell, P., and Zebulum, R. S. (1999). Explorations in Design Space: Unconventional electronics design through artificial evolution. *IEEE Transactions on Evolutionary Computation*, 3(3):167–196.

Tufte, G. (2009). The Discrete Dynamics of Developmental Systems. In *Proc. of 2009 International Conference on Evolutionary Computation (CEC 2009)*, pages 2209–2216. IEEE.

Turing, A. M. (1937). On Computable Numbers, with an Application to the Entscheidungsproblem. In *Proceedings of the London Mathematical Society 1936-37*, volume 42 of *2*, pages 230–265. London Mathematical Society.

von Neumann, J. (1993). First draft of a report on the edvac. *IEEE Annals of the History of Computing*, 15(4):27–75.

Wolfram, S. (1984). Universality and Complexity in Cellular Automata. *Physica D*, 10(1-2):1–35.

6.6 Paper P6

Evolution-in-Materio of a dynamical system with dynamical structures

Stefano Nichele, Johannes Høydahl Jensen, Dragana Laketić, Odd Rune Lykkebø and Gunnar Tufte

International Journal On Advances in Systems and Measurements, Volume 9

IARIA 2016

Dynamics in Carbon Nanotubes for In-Materio Computation

Stefano Nichele, Johannes Høydahl Jensen, Dragana Laketić, Odd Rune Lykkebø and Gunnar Tufte

Department of Computer and Information Science
Norwegian University of Science and Technology
Trondheim, Norway
Email: {nichele, johannj, draganal, lykkebo, gunnart}@idi.ntnu.no

***Abstract*—In-materio computation exploits physical properties of materials as substrates for computation. Evolution-In-Materio (EIM) uses evolutionary search algorithms to find such configurations of the material for which material physics yields desired computation. New unconventional materials have been recently investigated as potential computational mediums. Such materials may intrinsically possess rich physical properties, which may allow a wide variety of dynamics. However, how to access such properties and exploit them to carry out a wanted computation is still an open question. This article explores the dynamics in one particular type of nanomaterials which is used to solve computational tasks. Nanocomposites of Single-Walled Carbon Nanotubes (SWCNTs) and PolyButyl MethAcrylate (PBMA) are configured so as to undergo evolutionary processes with the goal of performing certain computational tasks. Early experiments showed that rich dynamics may be achieved, which may yield complex computations. Some indications of chaotic behavior were observed so further work was carried out with the aim of examining the dynamics achievable by such nanocomposites. Since it is not an easy task to access the physics at the very bottom of the material, investigation of the material dynamics is kept within the limits imposed by our measurement equipment and the level of observability enabled by it. Presented results show that interesting, complex dynamics is achievable by examined nanocomposites and that it depends on the type of signals used for the material configuration as well as on the material intrinsic properties such as percentage of SWCNTs in the nanocomposite.**

Keywords–Computation-in-Materio; Evolution-in-Materio; Evolvable Hardware; Carbon Nanotubes; Dynamical Systems; Complexity.

I. INTRODUCTION

Computations result from perturbations of some dynamical system. The observable output of the system is the result of its dynamics. Dependent on the type of dynamics exhibited by the system, computations of various complexity levels may be achieved. The type of dynamics depends on the physics of the system and on the way in which the system is manipulated. Our work considers novel nanoscale materials [1] and was carried out within the EU-funded NASCENCE (NAnoSCale Engineering for Novel Computation using Evolution) project [2]. The nanomaterials investigated within the project are nanocomposites of Single-Walled Carbon Nanotubes (SWCNTs) and polymer molecules (PBMA), and networks of gold nanoparticles. The investigation of nanocomposites is performed under the Evolution-In-Materio (EIM) scenario [3], [4].

EIM is a novel approach to designing computing devices where various materials are used as computational substrates. It is one approach that may emerge as an answer to the challenges of today's widely accepted semiconductor technology. Digital computers based on silicon technology are designed using a conventional top-down process by human engineers. Engineering of such processors poses technological challenges due to scaling down. Various design techniques are applied in order to sustain scaling down of the semiconductor technology but it is becoming increasingly difficult to fabricate transistors at the nanoscale.

This has motivated efforts towards novel technologies that will assume not only new computational substrates but also novel principles of the design of computing devices and their usage. EIM is a bottom-up approach in which the physics of a computing substrate is used to produce computations of interest. Different computational substrates have been previously explored such as liquid crystals and Field Programmable Gate Arrays (FPGAs) [5]–[7]. The configuration of the computing substrate, i.e., some material, undergoes evolutionary changes until some desired response of the material is achieved according to the computational task at hand. The digital computer accesses the material via a special board, which allows the Evolutionary Algorithm (EA) to apply configuration and input signals and read the material response which will guide the evolutionary search.

Figure 1 illustrates an EIM system. Three main entities can be distinguished: a digital computer, the material and the interface between the two. The system clearly shows the separation of an analog/physical domain in which materials operate and a digital domain in which the computer responsible for input/output mapping and configuration operates. In all such systems an interface is needed for bidirectional translation of signals between digital signals of the computer and analog signals in the physical domain of the material. As mentioned, the digital computer is used for running the EA, which generates a population of genomes, and translates each genome into suitable analog signals which can be sent to the interface board.

Further, the response of the material for a given configuration and input signals is translated from analog form as produced by the material to its corresponding digital value so that the computer can calculate the fitness value of the genome. The fitness value guides evolutionary search towards a solution to the problem at hand.

In order to produce interesting behavior under the EIM scenario, it is required that the material is able to exhibit rich dynamics. The richness of the exhibited dynamics can be attributed to the physical properties of the material. In a

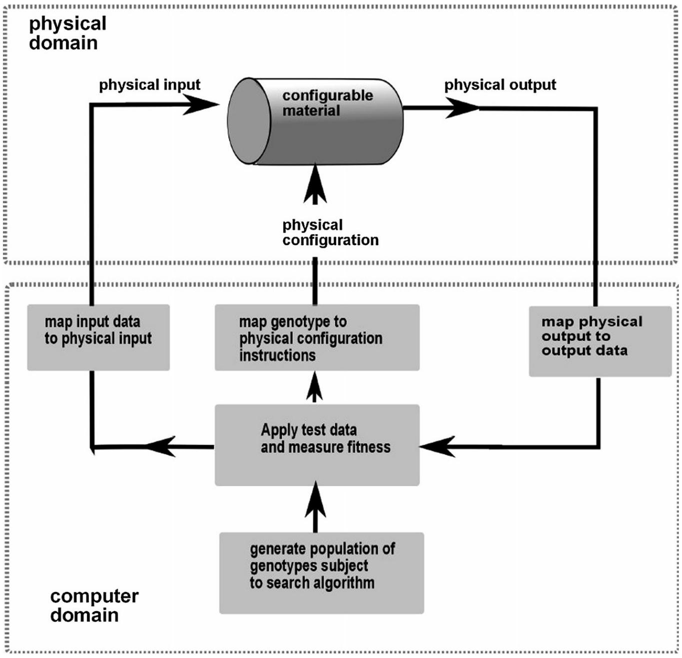


Figure 1. Principle of EIM illustrating the separation of an analog/physical domain where the material operates and the digital domain of computers, from [3].

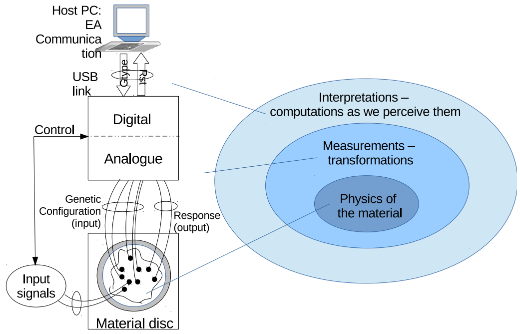


Figure 2. Conceptual domains of the computing system.

way it can be said that EIM manipulates the material so as to produce rich dynamics. The material blob is treated as a black box and EAs are used to "program" the material to solve a problem at hand.

Such a black box hybrid approach has been shown successful for a number of computational problems [8]–[13]. At the current state of research, it is not clearly understood what the exploited physical properties are and what the best way of exploring them is, e.g., what number of inputs and outputs and which types of signals - electrical (static voltages, sinusoidal waves, square waves) or even of some other kind such as temperature or light. The solved problems serve as a proof of concept that an EIM approach may be used for solving computational problems and indicates that it may be competitive in terms of computational time, size, and energy consumption. However, scaling-up to solve larger instances of a problem requires a better understanding of the dynamics exhibited by the material. In other words, the black box needs to be opened so that the underlying physical properties of the material are well understood. The number of used input electrodes, configuration signals available, etc. will directly affect the evolutionary search space.

Observing dynamics and its emergent complexity in computational materials is not an easy pursuit. Observability is limited by what output can be measured from the material and at which scale. At some scales we are not able to directly observe physical effects present in the materials, e.g., quantum effects due to mechanisms of electron transmission through carbon nanotubes. Therefore, we are limited to use signals which can be observed and measured. Figure 2 illustrates the taken approach to observe, exploit and gain an understanding of the dynamics of EIM systems. At the lowest level we have the physics of the material where computations happen, but due to nanoscale and even quantum effects, what is captured by our instruments will at best be just an approximation. In other words, the lowest level is inaccessible and must be treated as a black box. At the second level, the level of measurements and transformations, physical properties and dynamics are observable in the analog domain. This level can be explored to gain insight into the electrical properties of the material. The top level is the level of interpretations, i.e., computations as we perceive them. So, as shown in Figure 2, the dynamics of the analog signals are interpreted and transferred to data, i.e., the computational input – output mapping is performed. The top level is the level which is explored for computation. Here, it is important to note that the observations on the top level emerge as a result of all underlying dynamics.

The work presented in this paper includes a specific approach, as illustrated in Figure 2, to investigate the dynamics of the material at hand. The approach considers the complexity of the input - output mapping performed by the material for computation. Complexity is hard to measure even when well defined as, for example, Kolmogorov complexity [14]. Some approximations are needed if we want to obtain quantitative measures. In this work, we adopt *compressibility* as a measure of complexity.

This paper, which is an extended version of [1], is organized as follows: Section II provides background on EIM and position of the NASCENCE project within the field. Section III presents experimental platform Mecobo, which was developed within NASCENCE project, and which is used in our EIM experiments. Also, an experimental setup is explained as well as the material which was used in the experiments. Moreover, the section provides some background on different computational domains which can be distinguished under EIM computing scenario. Further, Section IV provides some initial results, presented in [1], which demonstrate interesting behaviors of the investigated material. Section V presents experiments which were conducted with the aim of investigating material dynamics in a greater detail. A measure of complexity is introduced which is used as the description of material behavior, the three sets of experiments are described followed by the results and the discussion which relates results to theoretical foundations. Finally, Section VI provides conclusion about the presented experiments and exhibited material dynamics within EIM computing.

II. Background – Evolution-in-Materio

The term Evolution-in-Materio was introduced by Julian Miller and Keith Downing in 2002 [4]. The general concept of EIM is that physical systems may intrinsically possess properties which may be exploited for computation.

A. Pioneering work

Early work on manipulation of physical systems for computation is related to the work of Gordon Pask [15], a classical cyberneticist whose pioneering work dates back to the 1950s. He tried to grow neural structures, dendritic wires, in a metal-salt solution by electrical stimulation [7]. His goal was to self-assemble a wiring structure within the material in order to carry out some sort of signal processing embedded in the material. He was able to alter the position and structure of the wiring filaments, and thus the behavior of the system. This was achieved by external influence, which consisted in applying different currents on electrodes in the metal-salt solution. This early version of material manipulation was done without aid of computers and different electrical configurations were tested manually. Stewart [16] later defined such a process as manufacturing logic "by the pound, using techniques more like those of a bakery than of an electronics factory".

B. Analog computers, FPGAs and liquid crystal

Later, Mills constructed an analog computer which he called Kirchhoff-Lukasiewicz Machine (KLM) [17]. The construction was done by connecting a conductive polymer material to logical units. The analog computation was carried out by placing current sources and current sinks into the conductive foam layer and reading the output from the logical units. One could argue that such machines were not easy to program due to the manual placement of connections into the material. On the other hand, some advantages of performing computation directly in the material substrate became obvious, e.g., a large number of partial differential equations were solved within nanoseconds.

In 1996, Thompson used intrinsic evolution to produce electrical circuits in FPGAs [5]. In his well-known experiment, he demonstrated that artificial evolution can be used to exploit physical properties of FPGAs to build working circuits, e.g., a frequency discriminator circuit. He found out that placing the circuit in a different part of the chip or disconnecting some unused modules would result in a non-working solution. Moreover, he was unable to replicate the chip behavior in simulation because evolution had exploited underlying physical properties of the FPGA. In fact, changing the FPGA with a similar model from the same producer would result in slightly different behavior. Thompson described such a process as "removing the digital design and letting evolution do it".

In [4], Miller and Downing suggested several materials which may be suitable for EIM, liquid crystals being among them. Simon Harding [18] later demonstrated that it was indeed possible to apply EIM on liquid crystals to evolve several computational devices: a tone discriminator [19], logic gates [20], and robot controllers [6]. Liquid crystal is a movable material where voltages affect orientation of the crystals. The movability was problematic since the material would undergo permanent changes during evolution. This led to unstable solutions that worked only once. Nevertheless, he showed that it was possible to quickly reach a working solution again by re-running the evolutionary algorithm for a couple of generations [19].

C. The NASCENCE project and recent work

Recently, the NASCENCE project [2] addressed nanomaterials and nanoparticles for EIM with the long term goal to build information processing devices exploiting such materials without the need to reproduce individual components. In particular, investigated nanomaterials included nanocomposites made of SWCNTs and polymer molecules and nanoparticle networks, in particular gold coated nanoparticles. Several hard-to-solve computational problems have been solved as proof of concept, e.g., Traveling Salesman [8], logic gates [9], bin packing [10], machine learning classification [11], frequency classification [12], function optimization [13] and robot controllers [21]. The SWCNT materials from the project are the subject of our investigation in this paper.

D. Interpretation and computation

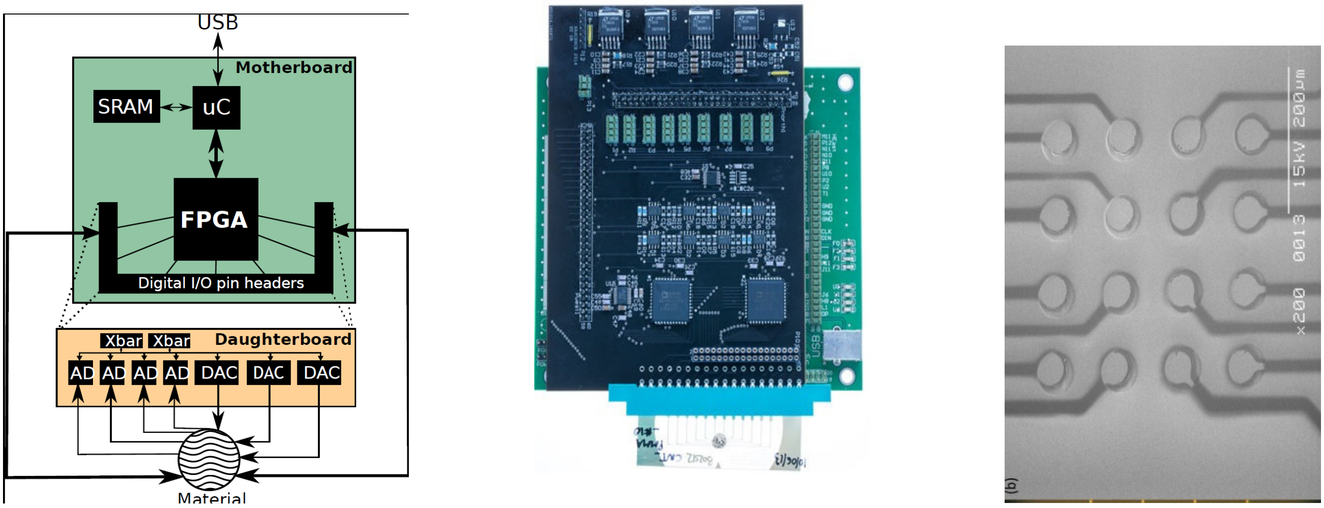
As stated, EIM has been used to solve a variety of problems. However, these results are all limited to a specific problem domain. To assess the potential computational power available in a material, we need a more general measurement. One way is to view *complexity* as indication of potential computational power [22].

Kolmogorov complexity [14], [23] is well-defined but incomputable in theory. However, it is possible to use measures such as compressibility to approximate complexity to some extent [24]–[27]. In fact, strings that are hardly compressible have a presumably high Kolmogorov complexity. Complexity is then proportional to the compression ratio.

High measurable complexity of output data or a high complexity ratio between input and output data may not always be a desired property. In classifier systems, such as Thompson's frequency discriminator [5], the output may be a binary response to a complex input signal. In this case the complexity ratio between output and input is very low. However, the computation, i.e., internal state transitions in the underlying physics of the material, is still a complex process but the complexity is unobservable since we only observe the input and output signals.

III. A platform for experiments and understanding of EIM systems

The conceptual idea of exploiting physics for computation requires a physical device, i.e., the material. In most EIM works, an intrinsic approach has been taken – computation is a result of real physical processes and the evaluation is a result of the performance of a physical system. An intrinsic approach allows access to all inherent physical properties of the material [3]. An analog computation [28] is a possibility, however, in this work a hybrid approach is taken. The hybrid approach includes the computational matter in a mixed signal system using a digital computer to configure and communicate with the material. Such an approach enables the computational power of the material with the ease of programmability of digital computers [2]. In a hybrid approach, observability is an issue, i.e., ensuring that the data from the material is observable and sound without using more computational power for the observation than the actual computation [29].



(a) Block diagram of the Mecobo hardware interface.

(b) Picture of the Mecobo motherboard with mixed signal daughter board.

(c) Electrode array, microscopic view. (Source: documentation of the NASCENCE project).

Figure 3. Overview of the Mecobo hardware interface.

A. NASCENCE's Mecobo: an experimental platform for EIM

A hybrid approach requires an interface between the digital world of computers and the analog world of materials. The Mecobo experimental platform [30] from the NASCENCE project is a hardware/software platform implementing the conceptual Evo-Materio-system shown in Figure 1.

Figure 3 shows an overview of the hardware interface: a Mecobo platform and microelectrode array on the material slide. A block diagram of the Mecobo platform is shown in Figure 3a. Configuration specification, i.e., genotypes, are loaded from a PC to Mecobo over a USB port. A micro-controller communicates with the USB interface and with an FPGA via an internal bus. The FPGA can be interfaced to the materials directly or, as shown in the figure, use a daughter board to extend the range of possible signals.

A picture of the Mecobo hardware is presented in Figure 3b. In the picture, the Mecobo is shown with a mixed signal daughter board and a material sample on a glass slide plugged in. Electrical connection between the material on the slide and the board is realized by the microelectrode array. A microscopic view of the microelectrode array before material disposition is shown in Figure 3c.

Mecobo is capable of controlling close to 100 individually configurable input/output signals (pins), which can be connected to the material. Each signal is described by parameters at a given point in time. For example, a pin can be programmed as a recording pin from time 0 to $100ms$, or as an output pin of square waves of some frequency from 0 to $1000ms$, or as an output pin of a constant voltage level, e.g., $2.7V$ from time 0 to $1500ms$ etc. Mecobo is connected to a host PC over USB and communicates via a Thrift server [31]. Communication based on Thrift technology also enables access to Mecobo remotely over the Internet. The maximum analog sampling frequency of the Mecobo board is $500kHz$. Input signals may be static voltages or periodic (e.g., square, sinusoidal) waves ranging in frequency between $400Hz$ and $25MHz$. For more details on Mecobo and an overview of the full range of programmable properties of the platform, see [30].

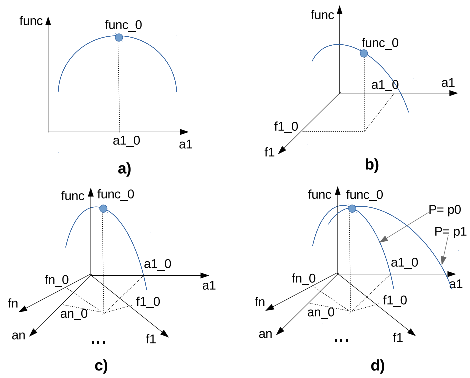


Figure 4. CNT computing system within a system theory framework.

B. Explaining computations within EIM

It can be said that computations are based on transformations of a system, so that the system input(s) and output(s) are related in some functional way. This functional relation can be expressed by a simple formula:

$$y = F(x) \tag{1}$$

where x and y correspond to an input and output of the system, respectively, and, in general, they are considered to be multidimensional and represented by vectors.

One way of analysis, more formally addressed within the system theory [32], [33] assumes that the system state is described by a set of variables that move through a state space.

For an EIM scenario, a better look into the state space of the system needs clarification of what is meant by system variables [34]. According to the explanation of different domains of computation as described in Section I, the variables of the system belong to the *domain of measurements* as schematically shown in Figure 2. The voltages and the set of properties which define them in this domain, i.e., amplitude, frequency and phase, can be represented with:

$$v_i = a_i \cdot func_p(f_i, \phi_i) \tag{2}$$

where v_i is voltage on the i-th electrode, a_i the amplitude, $func_p$ some periodic function, f_i frequency of the function $func_p$ and, finally, ϕ_i the phase of the voltage, all referring to the i-th electrode. The symbols are left lower case to remind that all of these values can be time varying.

Let us now consider an example in which for a system to perform functionality $func\_0$, for the input x_0, an output value y_0 is desired (Figure 4 a)). For simplicity, the variables on each of the axes are assumed to be scalars. When different configuration voltages are applied to the material, they change the system variables so that it passes through various states in the state space along some trajectory. Further, let us assume that only one electrode is used for configuration voltage and only one voltage parameter is changed, for example amplitude. By changing the amplitude along the a_1 axis different input-output mappings will be performed by the system. EIM would then search through the space until $func\_0$ point is reached. If also the frequency of the voltage v_1 is changed, then the state space could be searched along two axes as shown in Figure 4 b). And even further, if more than one electrode is used for configuring the material, then, in general, the space would look something like in Figure 4 c) and would be searchable along high number of axes, the limitation being only the physical number of electrodes in the system. Moreover, the state space may grow due to the change in some parameter, like temperature or light, as shown in Figure 4 d), which may all increase the size of the state space to search for the solution.

IV. A Detailed View of Material Dynamics

Experiments are performed on SWCNT mixed with PBMA on a micro electrode array supplied by Durham University. Material samples and micro electrode arrays are produced in a process where SWCNT-PBMA mixture is dissolved in anisole (methoxy benzene). The material samples are prepared on $4x4$ grids of gold micro-electrode arrays with pads of $50\mu m$ and pitch of $100\mu m$, see Figure 3c. The preparation is done by dispensing $20\mu L$ of the material onto the electrode area. The concentration of SWCNTs varies as shown in Table I where the material samples used in our experiments are listed. The SWCNT mixed with PBMA material dispersed over electrode array is baked for $30min$ at $90C°$. The solvent dries out and leaves a thick film of immovable SWCNTs supported by polymer molecules. The substrate is cooled slowly over a period of 1h. This process leaves a variable distribution of nanotubes across the electrodes. Typically, carbon nanotubes are 30% metallic and 70% semi-conducting, while PBMA creates insulation areas within nanotube networks. Such electrical properties of the material may allow non-linear current versus voltage characteristics.

The coverage of gold microelectrodes with randomly dispersed nanotubes varies and some of the electrodes may even be left with little or no coverage, as visible in the Scanning Electron Microscope (SEM) image in Figure 5.

TABLE I. Different materials used in the experiments.

Material	SWCNT Concentration, $wt\%$
B09S12	0.53%
B15S03	1.25%
B15S04	1.50%
B15S08	5.00%

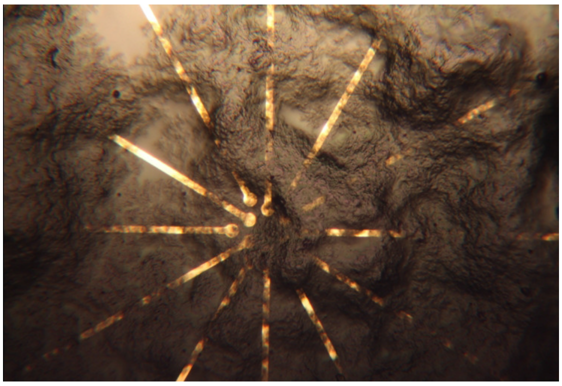


Figure 5. SEM image of gold electrode array with different coverage of nanotubes. Adopted from [9].

Initial investigation of the material response to various input signals showed several interesting behaviors in the material [1]. The goal was to gain insight into the material dynamics to identify suitable ways in which the material can be manipulated to perform computation.

As mentioned, EIM requires an interface between a digital computer which runs the EA and the material whose physics undergoes analog processes. This interface is typically provided by the Mecobo board. However, in order to better understand the underlying properties of the material and its responses, it is necessary to use more precise instruments. In these experiments, oscilloscopes and signal generators were used to get a more detailed view of the material dynamics.

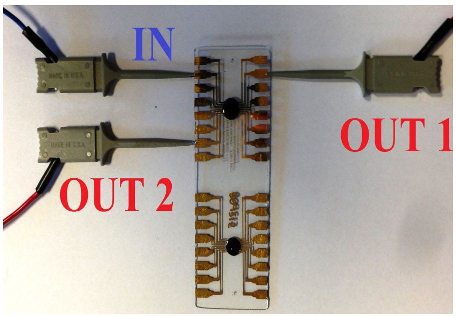


Figure 6. Material slide and pins connected to signal generator (IN) and oscilloscope (OUT).

A. Experimental Setup

In the experiments herein, we connect a material slide to a Hewlett Packard 33120A $15MHz$ function / arbitrary waveform generator (used as input) and an Agilent 54622D $100MHz$ mixed signal oscilloscope (used as output). Input signals are square waves at different frequencies from the signal generator and the output signals are recorded on the oscilloscope.

The input / output pins were chosen so that there would be an equal distance between microelectrode pads within the microelectrode array (Figure 3c). The placement of input and output signals on the material slide is shown in Figure 6, where the input probe (from the signal generator) is placed on pin #2 (IN) and the two output probes (to the oscilloscope) are connected to pins #9 (OUT1) and #7 (OUT 2).

B. Results and discussion

Figure 7 presents the experimental results. In particular, Figures a) show several snapshots of the material response on two different pins at different frequencies, ranging from $1KHz$ (Figure a1) to $14MHz$ (Figure a12). At $1KHz$ the signals may seem similar (a1), where the material charges-up and subsequently discharges, but in a zoomed in snapshot, i.e., where a part of the response is shown at a higher resolution (a2), a voltage spike is visible on the second probe which is not present on the first probe. This is better visible at $5KHz$ (a3), $30KHz$ (a4) and $100KHz$ (a5), where it is possible to notice that on the rising front there is a sudden voltage increase/drop. The material behavior is capacitor-like. Starting from $500KHz$ (a6), which is also zoomed in (a7), the second probe signal is similar to a square wave (most of the harmonic frequencies are passed) while the first probe acts more like a filter. The difference is caused by different concentrations of CNTs between the IN-OUT electrodes, i.e., different paths the current is enabled to follow between the electrodes. In both cases, there is a resonance phase which results in a deterministic yet semi-chaotic waveform. This may be the effect of some conducting sub-networks in the material that are enabled at specific frequencies and disabled at others. At 2, 5 and $8.5MHz$ the measured voltage decreases while frequency increases. At $10MHz$ (a11) a strange phenomenon is observed where both signals show a voltage increase. The effect is more prominent on the first output. We ascribe such behavior to be due to a feedback effect where harmonics of some frequencies are fed again into the material by some nanotube sub-networks. At $14MHz$ (a12) the signal on the second probe is sinusoidal, i.e., only one harmonic is present. As such, it may be concluded that with a single square wave input it is possible to observe a rich variety of behaviors while the frequency spectrum is traversed.

As the system produces uniform, stable, and semi-chaotic behaviors, it is of particular interest to visualize input-output responses and output-output relations in order to better understand traversed trajectories and attractors. For this purpose, XY plots are shown in Figure 7 b, where OUT1 is plotted against OUT2 and Figure 7 c, where IN is plotted against OUT1. In Figure 7 b1, some orbits are present at $30KHz$. Similar orbits are visible at $60KHz$ (b2) and $100KHz$ (b3), moving towards opposite corners to those where the impulse is. After each impulse, there is a semi-chaotic orbit that relaxes before the next impulse arrives, as the semi-chaotic behavior is annihilated by the lack of energy in the material, until the arrival of the next impulse. This suggests that chaotic behavior may be present, yet particularly difficult to observe.

XY plots between input and output are shown in Figure 7 c). These Figures represent the phase space of the system (input-output pin pair). Figure 7 c1 is obtained at $350Khz$. Several oscillating orbits are present, which are zoomed-in at $2MHz$ (c2). The same effect is observed for frequencies up to $5MHz$ (c3) while for frequencies around $10MHz$ and higher we observe a hysteresis loop, which indicates that some saturation may have been reached in the material. Some sort of non-linearity seems present, which is always a good indicator that the system may achieve complex behavior.

To summarize this set of results, even if a single square wave input signal is used, the resulting output shows a variety of behaviors. Square waves [35] produce richer dynamics than what may be achieved by a single static voltage or by a sinusoidal wave. Such richness of the response is due to the rich spectrum of the square waves which contains a variety of harmonics. In particular, some of the nanotube sub-networks may be sensitive to certain frequencies. Therefore, square waves may be better suited to penetrate the material and exploit the nanocomposite's intrinsic properties.

V. A Complexity View of Material Dynamics

The initial experiments with the oscilloscope measurements gave valuable insight into the different dynamics available in the material. However, such detailed measurements only give a very narrow view of the possible behaviors of the system. In order to get a broad picture of the space of possible material dynamics, one has to sacrifice some amount of detail. By using the Mecobo hardware platform (Section III) we are able to explore material dynamics at a higher level.

Mecobo allows us to scan a much wider range of signal frequencies, explore a myriad of different material locations and easily analyze the results on a PC. For these experiments we use the digital signal generator on Mecobo to generate square waves as input signals. The output signal is sampled as analog voltage using the on-board AD converter (Figure 3).

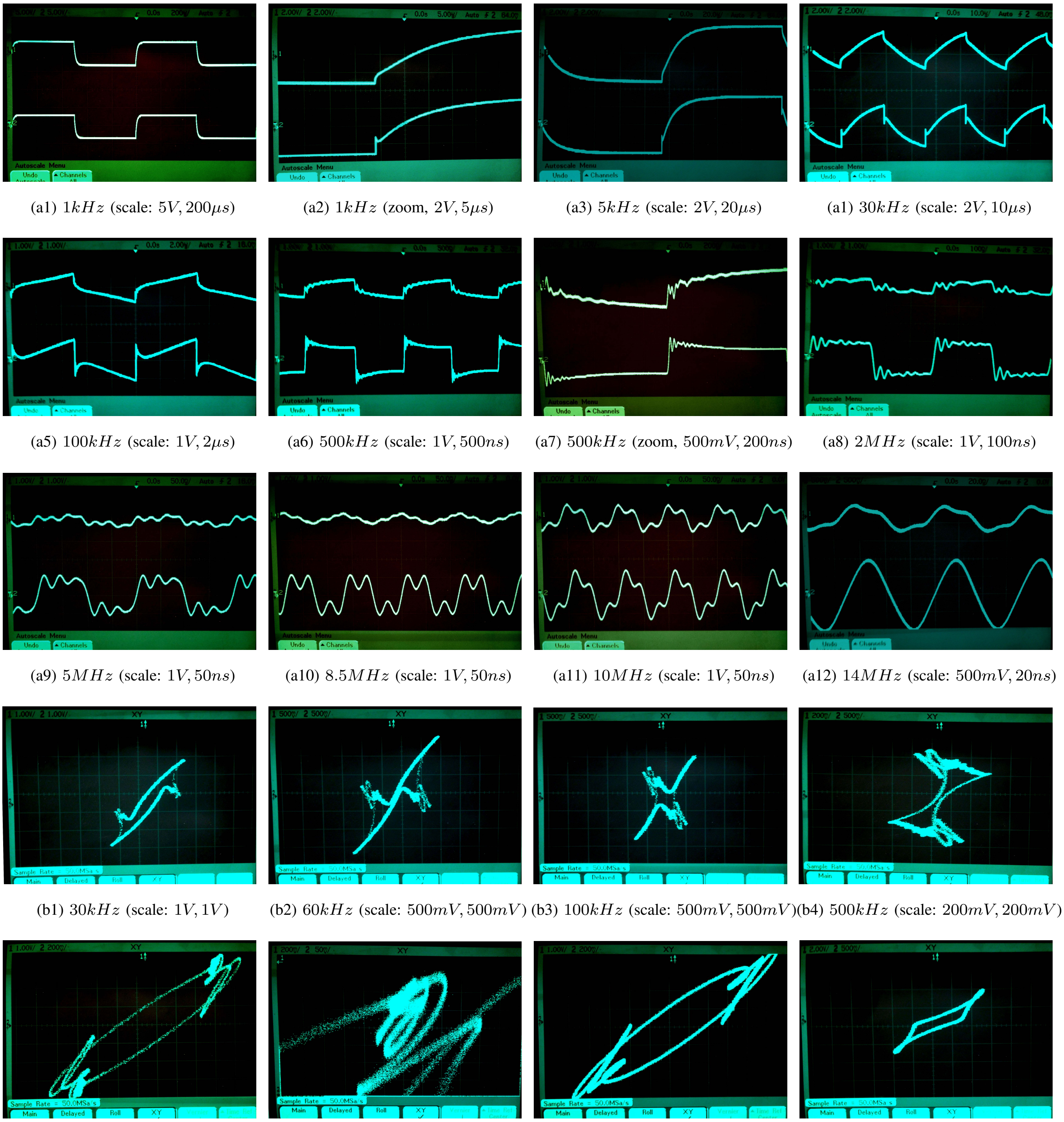
Complexity of the input/output signal is used as metric to classify different types of material dynamics. We use compressibility as an estimate of complexity as described in Section II-D. Since we are primarily interested in the complexity contribution of the material (and not the complexity of the input signal itself), we adapt the *complexity ratio*:

$$C_r = \frac{C_o}{C_i}$$

where C_o is the complexity of the output signal and C_i is the complexity of the input signal.

We present three sets of experiments where the computational complexity of the material is explored:

1) Complexity as number of input signals are increased
2) Complexity as function of one input frequency
3) Complexity as function of two input frequencies



(a1) $1kHz$ (scale: $5V, 200\mu s$) (a2) $1kHz$ (zoom, $2V, 5\mu s$) (a3) $5kHz$ (scale: $2V, 20\mu s$) (a1) $30kHz$ (scale: $2V, 10\mu s$)

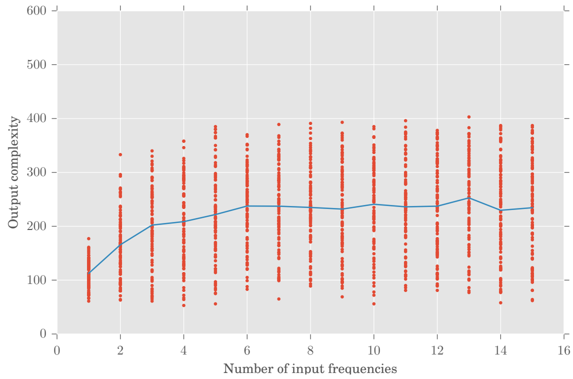
(a5) $100kHz$ (scale: $1V, 2\mu s$) (a6) $500kHz$ (scale: $1V, 500ns$) (a7) $500kHz$ (zoom, $500mV, 200ns$) (a8) $2MHz$ (scale: $1V, 100ns$)

(a9) $5MHz$ (scale: $1V, 50ns$) (a10) $8.5MHz$ (scale: $1V, 50ns$) (a11) $10MHz$ (scale: $1V, 50ns$) (a12) $14MHz$ (scale: $500mV, 20ns$)

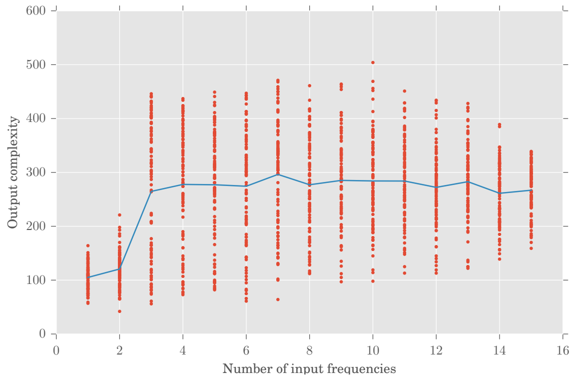
(b1) $30kHz$ (scale: $1V, 1V$) (b2) $60kHz$ (scale: $500mV, 500mV$) (b3) $100kHz$ (scale: $500mV, 500mV$) (b4) $500kHz$ (scale: $200mV, 200mV$)

(c1) $350kHz$ (scale: $1V, 1V$) (c2) $2MHz$ (zoom, $200mV, 50mV$) (c3) $5MHz$ (scale: $1V, 200mV$) (c4) $10MHz$ (scale: $2V, 500mV$)

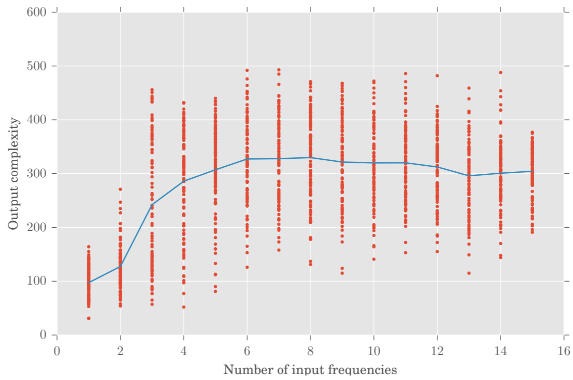
Figure 7. Oscilloscope screenshots. The resolution is indicated in parentheses. The resolutions have been chosen so as to be able to show interesting results at different scales.
(a) Voltage responses on 2 different pins with input square wave at different frequencies.
(b) XY plots, X (OUT1) is plotted against Y (OUT2) at different frequencies.
(c) XY plots, X (IN) is plotted against Y (OUT1) at different frequencies.



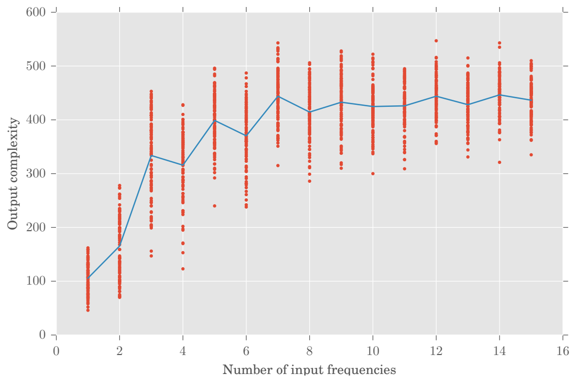
(a) Material B09S12 (0.53 wt% SWCNT)



(b) Material B15S03 (1.25 wt% SWCNT)



(c) Material B15S04 (1.50 wt% SWCNT)



(d) Material B15S08 (5.00 wt% SWCNT)

Figure 8. Output complexity as the number of input frequencies are increased from 1 to 15 for four different material samples. The red scatter plot shows individual measurements while the blue line indicates the mean values for each of the 100 data points.

A. *Experimental Setup*

For all the experiments, a set of input signals are sent through the material and a single output signal is recorded. The input signals are digital square waves in the range $400Hz$ to $25kHz$. The amplitude of the square waves is $0 - 3.3V$, which means that the material is exposed to a sharp rise and fall of the signal in this range. The duty cycle is held constant at 50%.

The output signal is recorded as analog voltage over time and sampled at a frequency of $500kHz$ for 10ms resulting in an output buffer of 5000 samples.

In order to compare the analog output signal to the digital input signal, we digitize the output signal by using the mean voltage as digital threshold. In other words, samples above the mean correspond to logical 1 and samples below the mean correspond to logical 0. To reduce sensitivity to noise, we apply hysteresis so that transitions between logic levels happen only if the analog voltage crosses the mean by a noise margin.

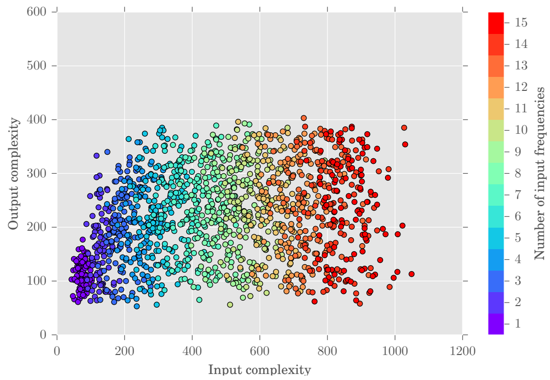
Complexity is estimated by compressing the sample buffer with zlib (zlib is based on LZ77 [36]) and calculating the length of the compressed string. Input complexity C_i is calculated based on a set of ideal square waves sampled at the same frequency as the output signal ($500kHz$).

All the experiments are repeated for the different material samples listed in Table I.

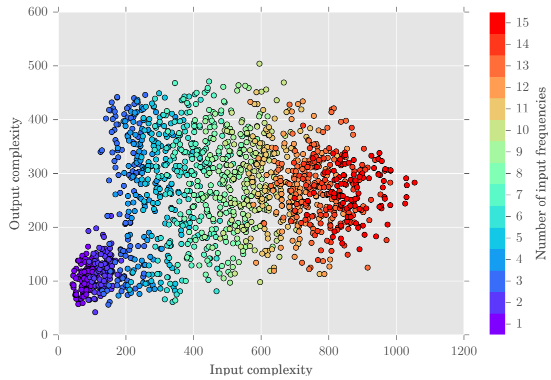
1) Complexity as number of input signals are increased: In the first experiment, the number of input pins are increased from 1 to 15. Input pins are selected at random and for each input pin a random frequency is chosen in the range of $400Hz - 25kHz$. The output signal is recorded from pin #0. The experiment is repeated 100 times for each number of input pins resulting in 1500 output signals.

2) Complexity as function of one input frequency: The second set of experiments provides a more detailed view of a subset of the first experiment by traversing the input frequency spectrum. Frequencies are increased from $400Hz - 25kHz$ in steps of $1000Hz$ resulting in 25 different input frequencies. The number of input pins are again increased from 1 to 15 but the same frequency is now applied to all input pins. In addition, both input pins and output pins are selected at random. For each number of input pins and for each frequency, the experiment is repeated 100 times resulting in 37500 output signals.

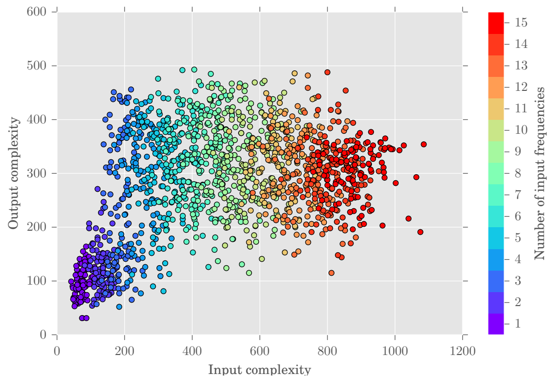
3) Complexity as function of two input frequencies: In the third experiment, we again traverse the same input frequency spectrum ($400Hz - 25kHz$), but this time for two input pins. In other words, the frequency spectrum is traversed in



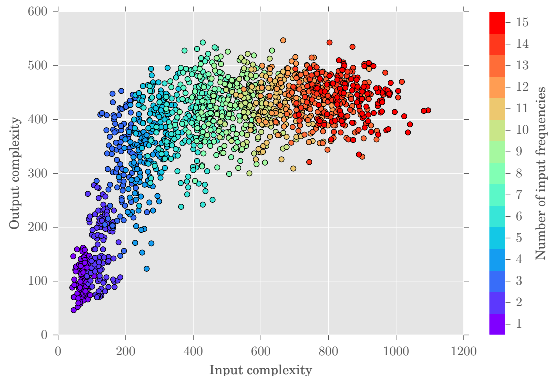
(a) Material B09S12 (0.53 wt% SWCNT)



(b) Material B15S03 (1.25 wt% SWCNT)

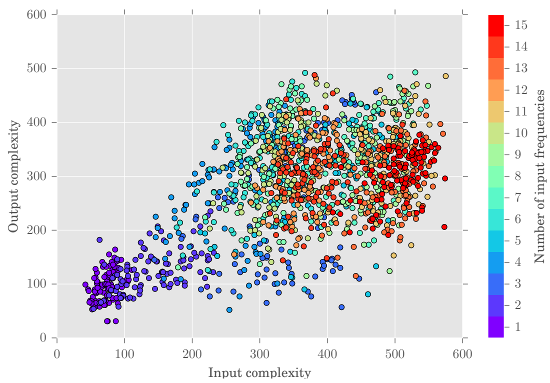


(c) Material B15S04 (1.50 wt% SWCNT)

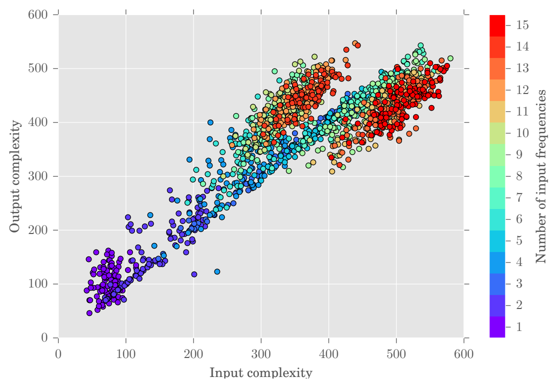


(d) Material B15S08 (5.00 wt% SWCNT)

Figure 9. Input vs output complexity as number of input frequencies are increased from 1 to 15 for four different material samples. The dots are colored according to the number of input frequencies used.

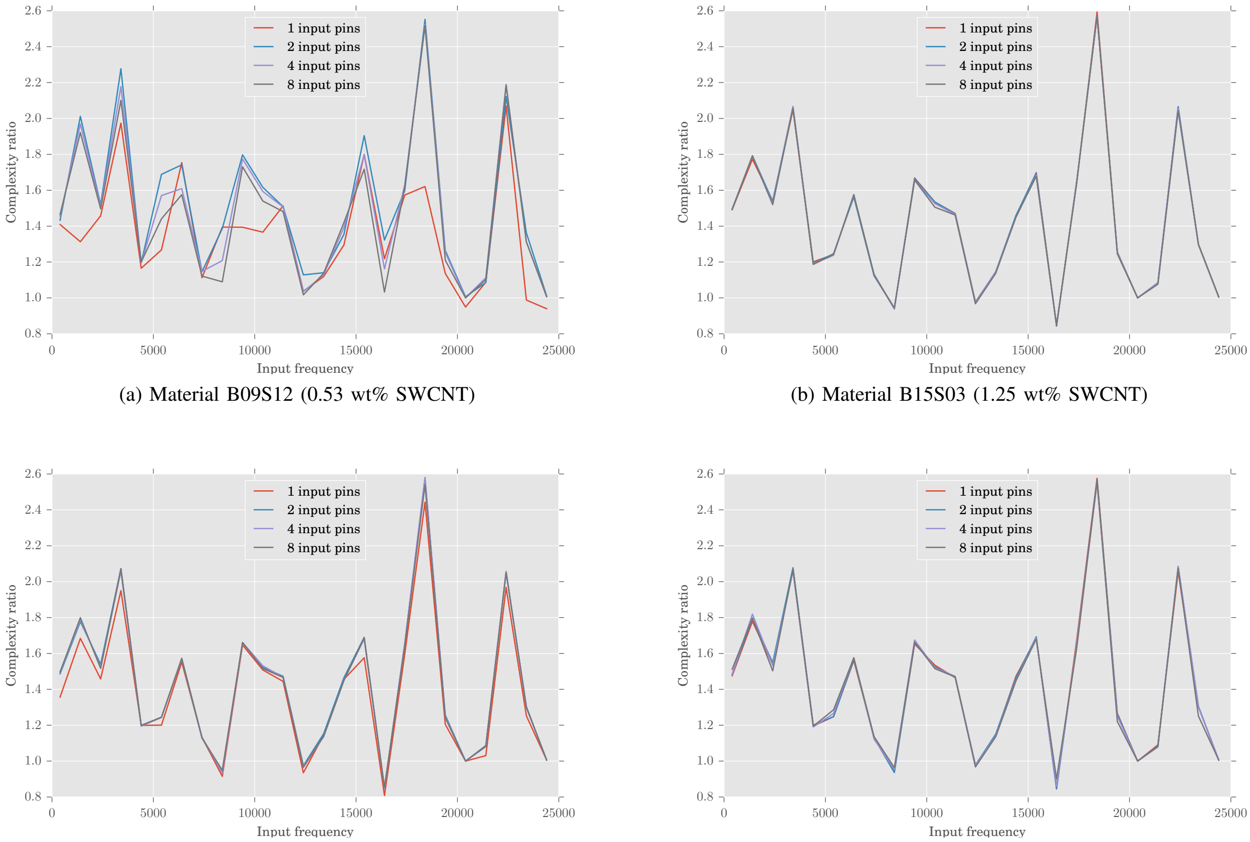


(a) Material B15S04 (1.50 wt% SWCNT)



(b) Material B15S08 (5.00 wt% SWCNT)

Figure 10. Input vs output complexity when the input signals are summed together before input complexity is estimated. Results from two material samples with different SWCNT concentrations are shown. The dots are colored according to the number of input frequencies used.



(a) Material B09S12 (0.53 wt% SWCNT)

(b) Material B15S03 (1.25 wt% SWCNT)

(c) Material B15S04 (1.50 wt% SWCNT)

(d) Material B15S08 (5.00 wt% SWCNT)

Figure 11. Mean complexity ratio as function of input frequency for 1, 2, 4 and 8 input pins. The same frequency is applied to all input pins. Results from four different material samples are shown.

two dimensions resulting in 25^2 pairs of input frequencies. Both input pins and output pins are selected at random. The experiment is repeated 10 times for each set of input/output pins.

B. Results and discussion

1) Complexity as number of input signals are increased: Figure 8 shows output complexity C_o measured over the range of 1-15 input frequencies. The blue line shows the mean output complexity value for each of the 100 data points. As shown in the plots, the output complexity increases with the number of input signals. There appears to be a fairly sharp rise in complexity as the number of square wave inputs are increased from 1 to 4. After this point the output complexity appears to saturate.

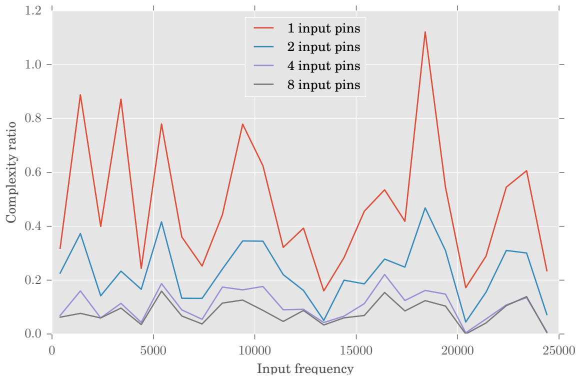
The scatter plot shows a fairly high variation in output complexity when the number of input signals exceeds one. This indicates that the materials exhibit a rich variety in output depending on the frequency and/or the choice of input pins.

A more detailed view is obtained when output complexity is plotted against input complexity (Figure 9). In these plots, it becomes clear that the input complexity C_i increases almost linearly with the number of input signals. Output complexity, however, saturates quickly above 3-4 input signals. In other words, above this level the added complexity from the input signal is not observed at the output.

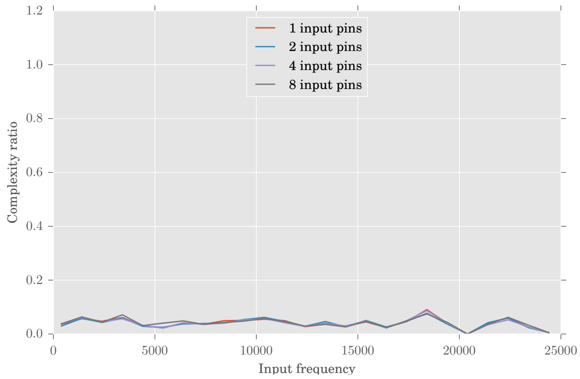
Again the richness of output complexity can be observed. The output signal is generally less complex than the input signal, which indicates that the material acts as a filter or stable attractor. However, there are situations where the complexity of the output signal exceeds that of the input signal. The input complexity is estimated from *ideal* square waves, which are not directly comparable to the signals generated by the hardware platform. However, the estimate does give an indication that the materials exhibit rich dynamics.

From Figures 8 and 9 it appears as if higher concentrations of SWCNTs result in higher output complexity. Such a trend is counter-intuitive, since as concentration increases the electrical resistance of the material is reduced. As resistance goes towards zero the material should act more like a wire, which means that the input signals should pass through unaltered. If multiple input signals are sent through a wire, the output signal would simply be the sum of the input signals. Therefore, it would be interesting to investigate how closely the output signal resembles the sum of the input signals.

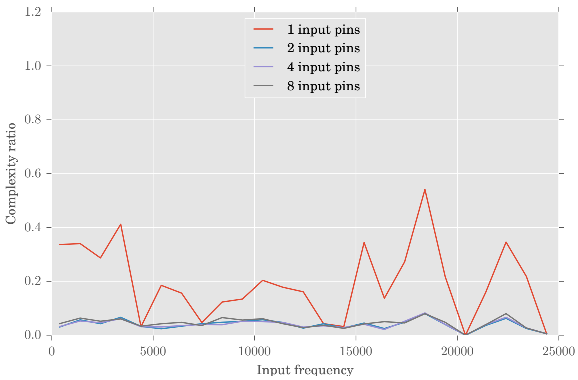
Figure 10 plots input vs output complexity when the input signals are summed together before C_i is estimated. For the material with high SWCNT concentration (B15S08, Figure



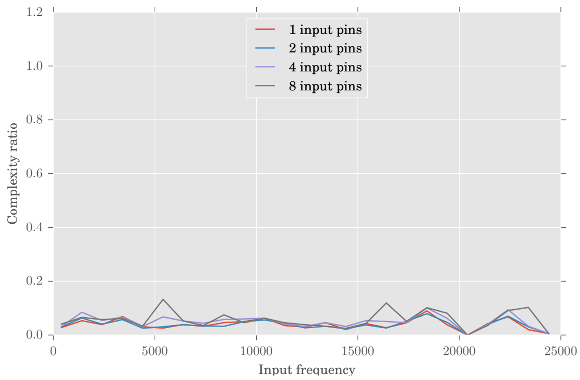
(a) Material B09S12 (0.53 wt% SWCNT)



(b) Material B15S03 (1.25 wt% SWCNT)



(c) Material B15S04 (1.50 wt% SWCNT)



(d) Material B15S08 (5.00 wt% SWCNT)

Figure 12. Standard deviation of complexity ratio as function of input frequency for 1, 2, 4 and 8 input pins. The same frequency is applied to all input pins. Results from four different material samples are shown.

10b) there is now a clear linear relationship between input complexity and output complexity. In other words, this material appears to behave much like a wire that simply sums the input signals together in some way. Lower SWCNT concentrations, however, display more diverse behavior as can be seen in Figure 10a, where there is no clear linear relationship between C_i and C_o.

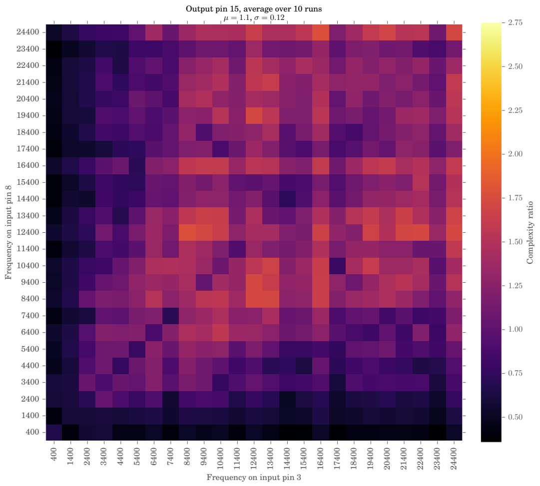
2) Complexity as function of one input frequency: Figure 11 shows the mean *complexity ratio* C_r over the range of input frequencies applied to the four material samples. From the plots it is evident that C_r is highly dependent on the input frequency with spikes at certain frequencies. Complexity appears to be fairly consistent across the four material samples, i.e., the materials are sensitive to the same frequencies.

Applying the input frequency to more pins does not seem to affect the mean complexity by much. However, there is a clear reduction in complexity variation, as can be seen from Figure 12, where standard deviation of the complexity ratio is shown. One possible explanation is that the input signal is effectively amplified as it is applied to more input pins.

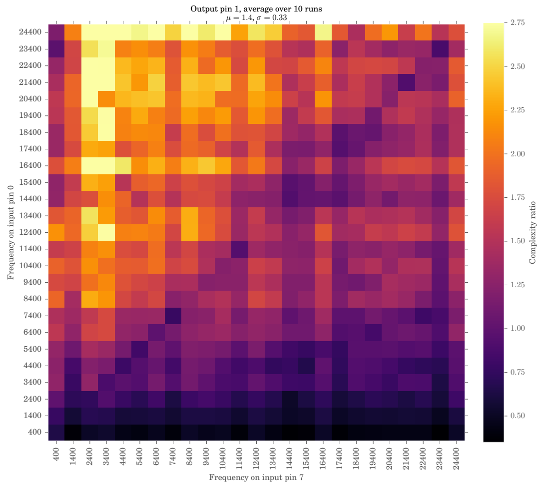
Another trend that can be seen from the plots in Figure 12 is an inverse relationship between complexity variation and the SWCNT concentration, i.e., more uniform output complexity with increased SWCNT concentration. This may be due to the fact that higher SWCNT concentration leads to a lower electrical resistance in the material and thus more pathways for the input signal to reach the output pin. However, one exception can be observed for the B15S04 sample where a higher variation is found when the frequency is applied to only one input pin. This likely indicates that one electrode is only partially connected to the material in this particular sample.

3) Complexity as function of two input frequencies: By sweeping the two input frequencies applied to the material we get a more detailed view of some of the results from the first experiment. Figure 13 depicts complexity ratio as a heat map where the two input frequencies are swept in the X and Y axes and color represents complexity. The colors range from dark purple (low complexity) to bright yellow (high complexity). As with one input signal, the heat maps show clearly that the complexity landscape is dependent on the selection of input frequencies.

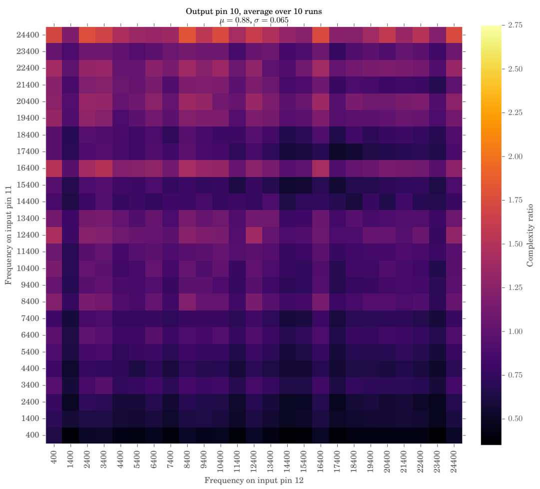
Figures 13a and 13b depict complexity for the same material sample B09S12, but with different selection of input and output pins. As can be seen, the two heat maps display clear differences in complexity ratio, where the latter pin configuration (13b) generally exhibits more complex output. However, this is not always the case, as can be seen in Figures 13c and 13d, where different input locations result in quite



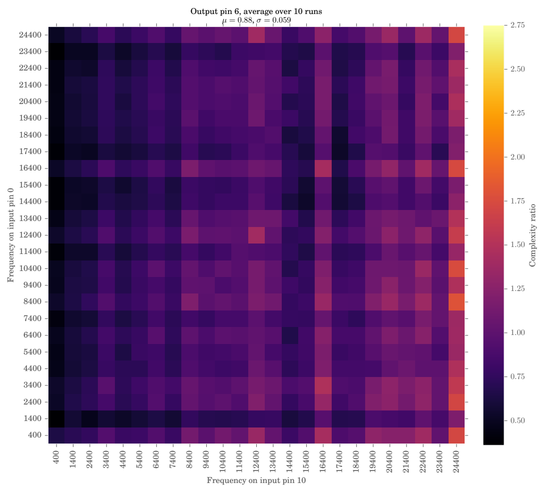
(a) Material B09S12 (0.53 wt% SWCNT), input pins 3 and 8, output pin 15



(b) Material B09S12 (0.53 wt% SWCNT), input pins 7 and 0, output pin 1



(c) Material B15S08 (5.00 wt% SWCNT), input pins 12 and 11, output pin 10



(d) Material B15S08 (5.00 wt% SWCNT), input pins 10 and 0, output pin 6

Figure 13. Complexity ratio as function of two input frequencies (X and Y axes). The heat maps shows complexity ratio C_r averaged over 10 runs. Colors range from dark purple (low complexity) to bright yellow (high complexity). Four heat maps are shown for two material samples: B09S12 (13a-13b) and B15S08 (13c-13d). Each heat map shows complexity when input is applied to different input/output pins.

similar complexity landscapes.

VI. Conclusion

The general ideas, experiments, and results presented relates to dynamics performed by SWCNT and PBMA nanocomposites, which may be exploited by EIM. The materials and experimental system has, as presented in Section II, shown promising computational behavior on a variety of problems. In this work, the behaviors are related to measurable dynamic behavior. That is, the experiments are designed to capture dynamic properties of the materials as to gain an understanding of what inherent dynamics are observable in an EIM setting. The approach taken is to view the material, i.e., physical system, as a hierarchical information processing device (Figure 2). At the bottom level the physical dynamics, i.e., quantum effects due to mechanisms of electron transmission through carbon nanotubes, are not observable within a reasonable resource usage. As such, the lowest level is treated only at a conceptual level. Dynamics at the bottom level are only observed as resulting voltages in the analogue domain. The information available at this level is exploited to gain insight into the electrical properties of the material when exposed to dynamic input stimuli. At the top level the material is interpreted as a discrete dynamical system. However, the observable dynamics at this discrete level is a result of all the underlying physics.

As stated by Miller et al. [3]: "...exploit the intrinsic properties of materials, or "computational mediums", to do computation, where neither the structure nor computational properties of the material needs to be known in advance". The statement may indicate that any material can be looked at as a black-box. However, insight into what properties are available for evolution provides knowledge on how to construct a successful EIM system. Our findings show that the materials exhibit rich dynamical properties observable at the analogue level. Figure 7 shows the behavior at an (close to) analogue time and voltage scale. The properties of these behaviors are available for exploitation by evolution, even if not explicitly controllable from the top discrete digital domain.

At the top level, the abstract measurements of complexity shows how such a measurement can indicate what computational problems the EIM system may handle. Especially the experimental results from Figure 13 show that the materials tested include behavior found in classifier systems, such as Thompson's frequency discriminator [5] (generally a trend of reduced complexity as illustrated in Figure 13d). From the same experiment, Figure 13b shows an increase in complexity generated by the dynamics of the material. A clear indication of a system which has more internal (observable) states than of the input data.

Our results also reveal several specific properties of the SWCNT materials used. In particular, as the number of input signals grows, a saturation of output complexity is reached. From an EIM perspective this is interesting, since it implies that information is filtered when many input signals are applied. The results also show a wide variety in output complexity depending on input frequency and selection of input/output pins. An indication that the materials are capable of many different modes of operation.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP/2007-2013) under grant agreement number 317662.

References

[1] S. Nichele, D. Laketić, O. R. Lykkebø, and G. Tufte, "Is there chaos in blobs of carbon nanotubes used to perform computation?" in FUTURE COMPUTING 2015, The Seventh International Conference on Future Computational Technologies and Applications. ThinkMind, 2015, pp. 12–17.

[2] H. Broersma, F. Gomez, J. Miller, M. Petty, and G. Tufte, "Nascence project: nanoscale engineering for novel computation using evolution," International journal of unconventional computing, vol. 8, no. 4, 2012, pp. 313–317.

[3] J. Miller, S. Harding, and G. Tufte, "Evolution-in-materio: evolving computation in materials," Evolutionary Intelligence, vol. 7, no. 1, 2014, pp. 49–67.

[4] J. Miller and K. Downing, "Evolution in materio: Looking beyond the silicon box," in The 2002 NASA/DoD Conference on Evolvable Hardware, A. Stoica, J. Lohn, R. Katz, D. Keymeulen, and R. S. Zebulum, Eds., Jet Propulsion Laboratory, California Institute of Technology. Alexandria, Virginia: IEEE Computer Society, 15-18 July 2002, pp. 167–176.

[5] A. Thompson, Hardware evolution - automatic design of electronic circuits in reconfigurable hardware by artificial evolution. CPHC/BCS distinguished dissertations, 1998.

[6] S. Harding and J. Miller, "Evolution in materio : A real-time robot controller in liquid crystal," in Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware, J. Lohn, D. Gwaltney, G. Hornby, R. Zebulum, D. Keymeulen, and A. Stoica, Eds. Washington, DC, USA: IEEE Press, 29 June-1 July 2005, pp. 229–238.

[7] P. Cariani, "To evolve an ear: epistemological implications of Gordon Pask's electrochemical devices," Systems Research, vol. 10, no. 3, 1993, pp. 19–33.

[8] K. Clegg, J. Miller, K. Massey, and M. Petty, "Travelling salesman problem solved "in materio" by evolved carbon nanotube device," in Parallel Problem Solving from Nature - PPSN XIII, ser. Lecture Notes in Computer Science, T. Bartz-Beielstein, J. Branke, B. Filipic, and J. Smith, Eds. Springer International Publishing, 2014, vol. 8672, pp. 692–701.

[9] A. Kotsialos, K. 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., vol. 10, no. 5-6, September 2014, pp. 473–497.

[10] M. Mohid, J. Miller, S. Harding, G. Tufte, O. R. Lykkebø, K. Massey, and M. Petty, "Evolution-in-materio: Solving bin packing problems using materials," in The 2014 IEEE Conference on Evolvable Systems - ICES, IN PRESS. IEEE Computer Society, 2014.

[11] M. Mohid, J. Miller, S. Harding, G. Tufte, O. Lykkebø, M. Massey, and M. Petty, "Evolution-in-materio: Solving machine learning classification problems using materials," in Parallel Problem Solving from Nature PPSN XIII, ser. Lecture Notes in Computer Science, T. Bartz-Beielstein, J. Branke, B. Filipic, and J. Smith, Eds. Springer International Publishing, 2014, vol. 8672, pp. 721–730.

[12] M. Mohid, J. Miller, S. Harding, G. Tufte, O. R. Lykkebø, K. Massey, and M. Petty, "Evolution-in-materio: A frequency classifier using materials," in The 2014 IEEE Conference on Evolvable Systems - ICES, IN PRESS. IEEE Computer Society, 2014, pp. 46–53.

[13] M. Mohid, J. Miller, S. Harding, G. Tufte, O. Lykkebo, M. Massey, and M. Petty, "Evolution-in-materio: Solving function optimization problems using materials," in Computational Intelligence (UKCI), 2014 14th UK Workshop on, D. Neagu, M. Kiran, and P. Trundle, Eds. IEEE, September 2014, pp. 1–8.

[14] A. N. Kolmogorov, "Three approaches to the quantitative definition of information'," Problems of information transmission, vol. 1, no. 1, 1965, pp. 1–7.

[15] G. Pask, "Physical analogues to growth of a concept," Mechanisation of Thought Processes, 1959, pp. 877–922.

[16] R. Stewart, "Electrochemically active field-trainable pattern recognition systems," Systems Science and Cybernetics, IEEE Transactions on, vol. 5, no. 3, 1969, pp. 230–237.

[17] J. W. Mills, "Polymer processors," Technical Report TR580, Department of Computer Science, University of Indiana, Tech. Rep., 1995.

[18] S. L. Harding and J. F. Miller, "Evolution in materio: Computing with liquid crystal," Journal of Unconventional Computing, vol. 3, no. 4, 2007, pp. 243–257.

[19] S. Harding and J. F. Miller, "Evolution in materio: A tone discriminator in liquid crystal," in Evolutionary Computation, 2004. CEC2004. Congress on, vol. 2. IEEE, 2004, pp. 1800–1807.

[20] ——, "Evolution in materio: Evolving logic gates in liquid crystal," in In Proceedings of the workshop on unconventional computing at ECAL 2005 VIIIth European. Beckington, UK, 2005, pp. 133–149.

[21] M. Mohid and J. Miller, "Evolving robot controllers using carbon nanotubes," in The 2015 European Conference on Artificial Life. The MIT Press, 2015.

[22] T. Kowaliw, "Measures of complexity for artificial embryogeny," in Proceedings of the 10th annual conference on Genetic and evolutionary computation. ACM, 2008, pp. 843–850.

[23] M. Li and P. Vitányi, An introduction to Kolmogorov complexity and its applications. Springer Science & Business Media, 2013.

[24] S. Nichele and G. Tufte, "Measuring phenotypic structural complexity of artificial cellular organisms," in Innovations in Bio-inspired Computing and Applications. Springer, 2014, pp. 23–35.

[25] M. Hartmann, P. K. Lehre, and P. C. Haddow, "Evolved digital circuits and genome complexity," in Evolvable Hardware, 2005. Proceedings. 2005 NASA/DoD Conference on. IEEE, 2005, pp. 79–86.

[26] P. K. Lehre and P. C. Haddow, "Developmental mappings and phenotypic complexity." in IEEE Congress on Evolutionary Computation (1). Citeseer, 2003, pp. 62–68.

[27] H. Zenil and E. Villarreal-Zapata, "Asymptotic behavior and ratios of complexity in cellular automata," International Journal of Bifurcation and Chaos, vol. 23, no. 09, 2013, p. 1350159.

[28] B. J. MacLennan, "A review of analog computing. technical report ut-cs-07-601," University of Tennessee, Knoxville, Tech. Rep., 2007.

[29] H. J. Bremermann, Self-Organizing Systems-1962. Spartan Books, 1962, ch. Optimization through Evolution and Recombination, pp. 93–106.

[30] O. R. Lykkebø, S. Harding, G. Tufte, and J. Miller, "Mecobo: A hardware and software platform for in materio evolution," in Unconventional Computation and Natural Computation - 13th International Conference, UCNC 2014, London, ON, Canada, July 14-18, 2014, Proceedings, 2014, pp. 267–279.

[31] A. S. Foundation, "Apache thrift." [Online]. Available: https://thrift.apache.org/

[32] W. R. Ashby, Design for a Brain, the origin of adaptive behaviour. Chapman & Hall Ltd., 1960.

[33] L. von Bertalanffy, General System Theory. George Braziller, Inc., Revised edition, Fourth printing, 1973.

[34] D. Laketić, G. Tufte, O. R. Lykkebø, and S. Nichele, "An explanation of computation – collective electrodynamics in blobs of carbon nanotubes," in EAI, BICT 2015, 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). ACM Digital Library, 2015, pp. 1–6.

[35] O. R. Lykkebø, S. Nichele, and G. Tufte, "An investigation of square waves for evolution in carbon nanotubes material," Proceedings of the European Conference on Artificial Life 2015, 2015, p. 503510.

[36] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," IEEE Transactions on Information Theory, vol. 23, no. 3, May 1977, pp. 337–343.

Appendix A

The Mecobo Material Interface version 4.1

This is included in the thesis because it has been an important tool during writing and in the hopes that some lessons might be gleamed from the work put into the system.

The Mecobo material interface is a system for stimulating materials and observing the result of the stimulus. It consists of three distinct hardware parts, the motherboard, a daughterboard and a “host” computer.

A.1 Design goals

As discussed in chapter 3, the design goals of the system moved in parallel with us getting more experience with the material. Some goals were however always present:

- A client/server model allowing work across the internet.
- Electricity as the main method of manipulation.
- Low cost. Since we expected an iterative process, each iteration should be cheap enough to allow as many iterations as time allows.
- Possible to build in the computer hardware lab at NTNU.

A.2 Architecture

The overall principle, which is based on the music production tool called a *sequencer,* is sketched in Figure A.2.1. The boundary where the signal leaves and enters the system is simply called a “pin”. There is no requirement for this to be attached to a material probe, there might be other devices downstream attached to the pin, however as far as Mecobo is concerned the stimulus exits and enters through the pin. In current implementations the pin is a physical piece of copper that eventually leads into the material.

Each pin is controlled by a **Pin Controller**, or PinC for short, which is fed commands by the **Scheduler.** The scheduler is timed by a central 1MHz clock that is connected to all units in the system that has to run relative to a central time. The scheduler receives the items to be scheduled (simply called Mecobo command packs) from the **Server** which has a public communications interface allowing **Clients** to send items to it. Collection of data, observing and measuring happens through the pin controllers setup to input mode. The data is then fetched from the pin controllers by the **Sample Collector** and eventually find their way back through the server to the client.

Each **item** mentioned above comes with an assigned **pin,** start and stop time relative to the **Central Clock**, and other information depending on the item’s type. For example, a stimulus item includes the nature of the stimulus such as “Square Wave”, “Frequency”, “Peak-to-Peak Voltage” and “Duty Cycle”. A recording item on the other hand will include the sample rate and the recording method.

A.3 Implementation

The Mecobo version 4.1 is implemented on two custom PCBs made by Lykkebø and Tufte in the CARD group at the Institute of Computer Science and Information Science at NTNU: The motherboard and the daughterboard.

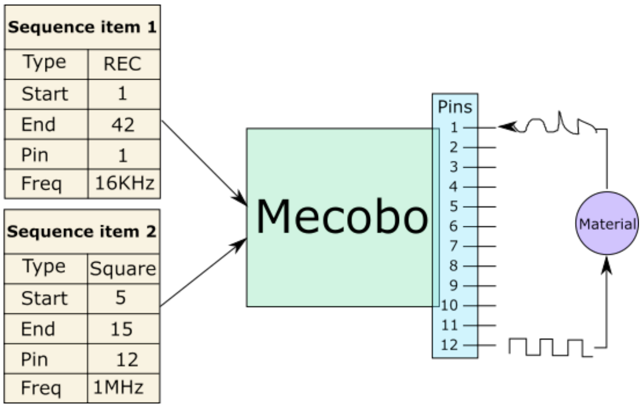


Figure A.2.1: The sequencer principle. A set of sequence items, scheduled to start and stop at designated times is sent to Mecobo, potentially over some network connection. In this case, one recording item (type REC) and one square wave output (type Square) is sent as two separate items which will be sequenced by Mecobo. The output from Mecobo is fed into the material on pin 12, and the observed state is recorded on pin 1.

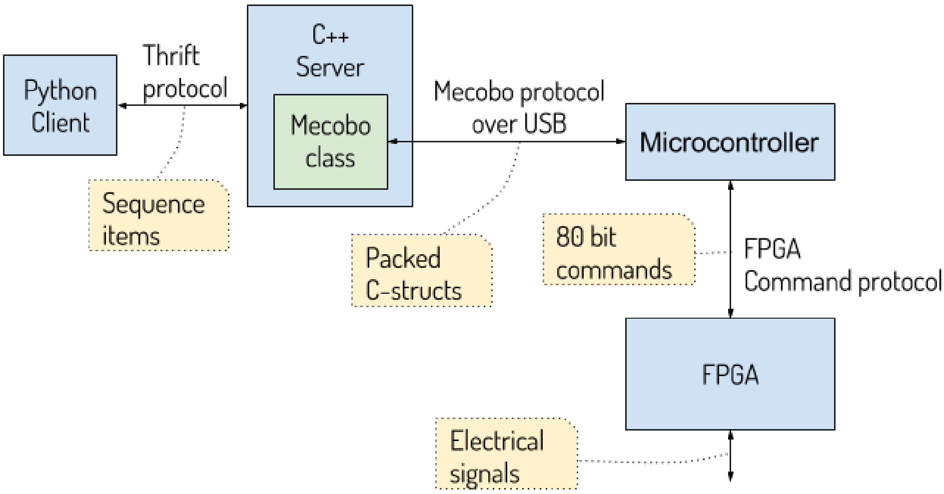


Figure A.3.1: Overview of the data flow in the Mecobo platform

The motherboard handles the host-board link, and has the capability to use the pins on the FPGA as an electrical interface towards a material. One can both sample (digitally) and output (digitally) with the Mecobo motherboard alone. If, however, one requires analogue signals over a greater voltage range, the daughterboard must be employed.

The motherboard design is split into two programmable parts; an FPGA and a Microcontroller, each with access to its own local memory used primarily for FIFOs. The microcontroller exposes an Ethernet connection and a USB connection, however the USB is the only interface currently in use. The USB connection connects the board to a host computer running a Thrift Server, exposing the interface to the board and handling the network communication with the clients. An overview of the data flow in the Mecobo system can be seen in Figure A.3.3.

A.3.1 PCB layout and component choice

PCB layout is a trade-off between keeping components accessible to debug probes, convenience in hand soldering and availability of components in required packages. We were fortunate enough in this project to have access to a reflow solder oven, allowing us to use BGA packages with far higher pin density than the quad flat packs that allow hand soldering. The down-side to the BGA packages is that it becomes hard to debug individual pins such as power and ground that are not routed to pin headers and such.

The power supply is done using linear regulators to keep noise as low as possible. Two adjustable regulators transform the 5V input voltage into two 1.2V and 3.3V domains.

We have used an 8-layer board stack to allow for efficient routing of the 384 pins of the FPGA package. All traces have been hand-routed, with traces on neighbouring layers routed perpendicular to each other to

FPGA	Xilinx Spartan 6LX
Microcontroller	Silicon Labs EFM3299GG
NOR flash	Cypress S29PL127J

Table A.1: Mecobo motherboard main components

Op-Amp	Analogue Devices 820
Cross bar	Analogue Devices 75019
DA-converter	Analogue Devices 5308
AD-converter	Analogue Devices 7327

Table A.2: Mecobo analogue extension board main components

minimize cross talk, and in addition copper fill on top and bottom layers have been via-stitched together in a further attempt to diminish noise.

The most important can be seen in Table A.1 and A.2.

A.3.2 FPGA Digital Design

The digital design was done in Verilog; with some verification done in System Verilog. The use of FPGA hardware design allows rapid development of units with strict control of timing, which was a requirement whose importance increased as the project developed.

At a glance The design is centred around a global clock running at 1MHz that controls all timing activities, giving a timing granularity of 1us. An overview of the system can be seen in figure A.3.3. The FPGA communicates with the outside world, in our case the microcontroller, via the External Bus Arbiter (EBA), which is a 16-bit wide data bus with 24 bits of address. The microcontroller sends 80 bits “commands” to the FPGA (see Table A.3). The ultimate destination of the commands are the Pin Controllers on the right side of Figure A.3.3. A command has per-unit context sensitivity, however they mostly either set up the output from the unit such as “10KHz square wave starting at time 142 with 30% duty cycle”.

The EBA collects 1 such command and enters it into the FIFO. The scheduler peeks at the bottom of the FIFO and compares the Global Clock to the START_TIME field, scheduling the command to the addressed unit. The scheduler runs at the system clock speed of 100MHz clock and uses 1 clock cycle per scheduling action. The scheduler runs 1 cycle ahead of the global clock time, and the commanded units will not change their internal state based on the command until the specified time has arrived. This enforces a strict serialization of the incoming commands, however it also imposes a limitation in terms of the number of simultaneous commands processed by the pin controllers. In practice, this does not seem to be an issue but the architecture might need more reworking to allow faster scheduling. An obvious “fix” is to move the In-FIFO into per-unit smaller FIFOs.

External Bus Arbiter The External Bus Arbiter (EBA) arbitrates the External Bus Interface (EBI) bus, connecting the microcontroller and the FPGA. A falling-edge detector on the write-enable line of the EBI informs the EBA that a write-transaction is finished. The EBA is a slave of the microcontroller who always acts as master, and will thus never drive the bus except if a bus-read is requested by the microcontroller.

Scheduler The scheduler is implemented as a state machine whose idle state peeks at the START_TIME of the *command* at the head of the FIFO. The address and data of the command are then forwarded to the command bus, and the write signal is asserted. The addressed controller then reads the bus.

Sample Collector The sample collector is a programmable unit that collects samples, or input, from the pin controllers and the AD controllers. The sample collector accepts commands that add the address of the sample-collecting controllers to a list that is iterated by a state machine. Each unit on this list has a “collect

32 bits: START_TIME	32 bits: DATA	16 bits: ADDRESS

Table A.3: The *COMMAND* format

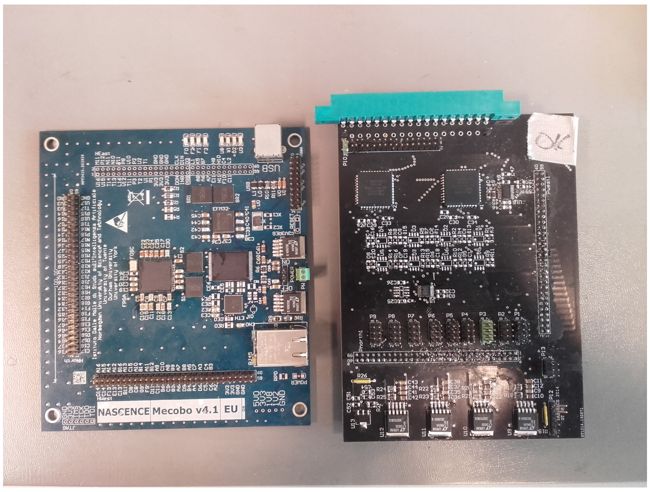


Figure A.3.2: The final version of the board, Mecobo version 4.1, along with the add-on daughterboard for electric analogue input and output.

sample" signal, instructing the unit to put its current collected sample on the *Sample Bus*. The sample data consists of 16 bits of samples and a 16 bit counter. The counter value is compared to the last collected sample by the sample collector, and if it differs, the sample is stored in the out-FIFO. In this way, the Sample Collector is constantly polling the units for new samples, but there is no need for the sample collector to know about the used sample rate, this is only a function of the specific controllers.

Pin Controller The pin controller's duty is two-fold:

- Output square waves of varying duty cycles, Pulse-Width modulated signals and Constant values.
- Implement "Digital Sampling", i.e. sampling the pin connected to the material at a certain rate.

These functions are realized through a state machine that idles until a command is received on the command bus interface. It then switches to either one of the output states, one for each mode of output (i.e. constant, square waves), or to a sample-collection state. If a new command is received on the command interface, the control logic will switch state in response.

The square wave output is implemented using "Direct Digital Synthesis". A "Phase Increment" register is programmed by the user. On each tick of some reference clock, another "Phase Accumulator" register has this phase increment added to it. At an average constant frequency, this leads to an overflow in the phase accumulator register. This overflow bit is directly connected to the output of the Pin Controller unit. PWM is currently not implemented. Sampling is implemented by simply updating a flip-flop register at the desired rate.

AD Controller The AD controller implements a serial wire interface towards an external AD chip. It is completely transparent in that it does not know about the particulars of the AD chip. After receiving a command on the command bus, the controller loads a 16 bit shift register and enters a busy-state-loop clocking out the bits serially. After this is finished, the state machine returns to the sample-collecting state, where it loads a 16 bit shift register with the input serial line and stores this as a collected sample.

DA Controller The DA controller is similar to the AD controller, in that it acts as a transparent serializing unit, taking in a 16-bit word on the command bus and clocking this out on the serial output line of the DA before returning to its idle state, waiting for a new command.

A.3.3 Software

The software written for the Mecobo system consists of three parts:

- A client, written by the user of the board, running on a computer somewhere in the world.
- A server, written by us, that listens for incoming connections and runs on a *host* computer.
- Micro controller code, running on the Mecobo board that communicates over USB with the server.

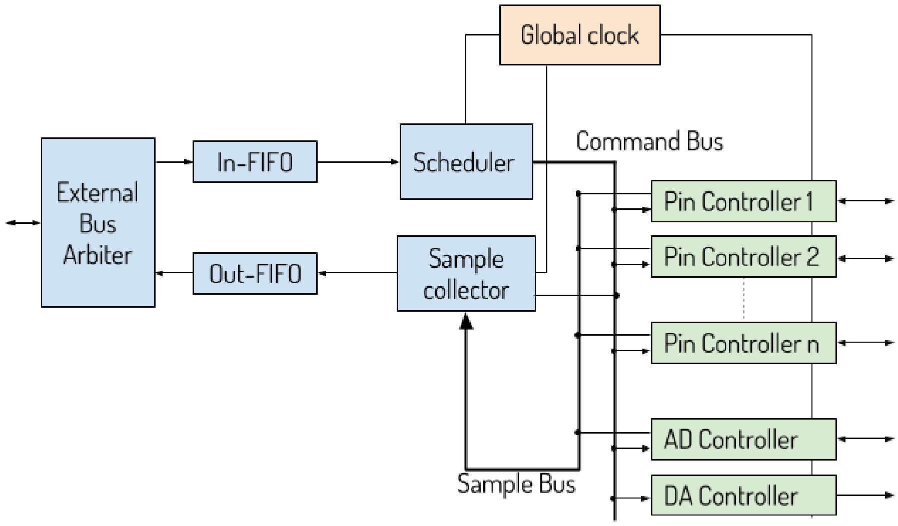


Figure A.3.3: Overview of the FPGA hardware

The client relates to Mecobo through a set of "Sequence Items" along with some special remote procedure calls, which are high-level descriptions of the wanted signal:

- Pin
- Output or Input pin (called signal or recording)
- Signal form, i.e. square waves and parameters on this signal
- Start time
- End time

A set of these sequence items make out a full sequence with a well-defined start and end time. These sequence items are received by the server that sorts them in time, and upon receipt of a special signal, "*runSequences"*, translates them into a special protocol that is a better fit for USB transfer onto the Microcontroller. The Microcontroller does further transformations into the format expected by the FPGA firmware.

Microcontroller software

The microcontroller handles the host-board interface, which currently is a USB interface. An additional responsibility is to load the programming data for the FPGA from the on-board NOR flash. The microcontroller also acts as a decoder and buffer for the commands sent over the USB interface. In addition, it buffers collected samples from the FPGA when it is not busy sending new data. This 4-buffer model (2 FIFOs on the microcontroller and 2 FIFOs on the FPGA) allows us to continuously stream configuration, input and observations back and forth between the material and the host computer. The FIFOs are implemented as thread-safe linked lists using atomics.

A main loop runs continuously, constantly polling the FPGA data FIFO for data. If there is more than a set threshold items, they are fetched in a batch mode fashion and stored in local SRAM on the FPGA. The FPGA access takes 6 cycles and a SRAM operation is done in 3, meaning that there is time left for a read operation to be interleaved. The USB driver sets up a DMA channel from the SRAM to fetch data when requested by the host computer.

Pins and channels

The incoming data from the FPGA is stored as 16-bit words in which the top 3 bits are used as channel identifiers, which are an important abstraction in the host and microcontroller code stemming from the crossbar switches on the Analogue Daughterboard. A *pin* is the endpoint of a recording or output item, and relates to the physical pins either on the Mecobo, or on the daughterboard if it connected. A *channel* is assigned to a pin in the host computer software, however this is transparent to the end-user (and, in fact the FPGA as well). The microcontroller holds a "pin-to-channel"-map (and its reverse) to do the mapping back and forth when passing data to and from the transparent layers.

Figure A.4.1: The infrared reflow oven used for soldering the ball grid array (BGA) components to the Mecobo motherboard.

When there is no daughterboard attached, the channels map directly to pins and the top 15 bits of the word becomes the pin identifier.

Host computer software

The host computer runs a server implementing a "Thrift" interface, which is a cross-platform language-independent client/server-interface description language along with code generating tools to generate the client and server code skeletons together with the communication code and protocols between the two. The server is implemented in C++ and accepts incoming connections from clients written in most common programming languages through a TCP socket.

The generated server skeleton code instantiates a singleton object of the "Mecobo" class, that implements the host-computer-to-board USB interface. When a "*runSequences*" command arrives, a new thread is spawned to handle the collection of data from the Mecobo. This allows us to run the data collection part arbitrarily long, provided that the bandwidth requirements between the FPGA and the host are not exceeded.

A.4 Practicalities

We had all our boards produced in China through a proxy company in Norway. This allowed for relatively cheap production, while at the same time avoiding interaction with the PCB factories. We had no problems with this approach, and only once did we encounter issues with board quality due to a mismatching silk print.

The board assembly was handled in the group's hardware lab in a time-consuming process. We eventually managed to spend only about 1 hour on producing a single platform, but the process was error-prone. Of particular use was the purchase of an infrared reflow oven, shown in A.4.1. This allowed us great freedom in the upper end of the component spectrum, which again allowed us to build more flexible platforms. Not having to solder 300 pins on a flat quad pack using a soldering iron was also a huge time saver.

One fundamental flaw with the board as it stands at the end of this project was the use of 2.54mm pin headers as the board-to-board connectors for expansion. For relatively small connectors of 20-30 pins, these work excellent, but for larger connectors, the friction generated when extracting and inserting the daughterboard was large enough to damage the boards by bending them.