Chapter 3
In materio computation using carbon nanotubes

Julian F. Miller and Simon Hickinbotham

All computing is physical (Landauer, 1996), requiring a substrate in which to perform the computation. The distinction between *in materio* computation and CCOMP is that the latter engineers logical states in the substrate that are then used as the basis for executing processes that obey the Church-Turing principle via the von Neumann architecture. By contrast, *in materio* computation is not aimed at imposing a model upon the substrate but rather with exploiting a naturally-occurring computational property that it may have.

This case study contains many quotations. Within quotations, double square brackets appearing within quotes indicate the reference numbers in the work from which the quote was taken. For example "[[3]]"(Miller et al., 2014) would indicate reference number 3 within the text of bibliography item (Miller et al., 2014).

3.1 Overview

3.1.1 In materio ***computing***

The goal here is to find out simultaneously both what the computational power of the substrate might be and also how best to program it. An evolutionary search-based algorithm is usually employed to "exploit physical characteristics within the material without much concern as to how those characteristics could be formally understood or engineered" (Clegg et al.,

Julian F. Miller
Department of Electronic Engineering, University of York, UK

Simon Hickinbotham
Department of Computer Science, University of York, UK

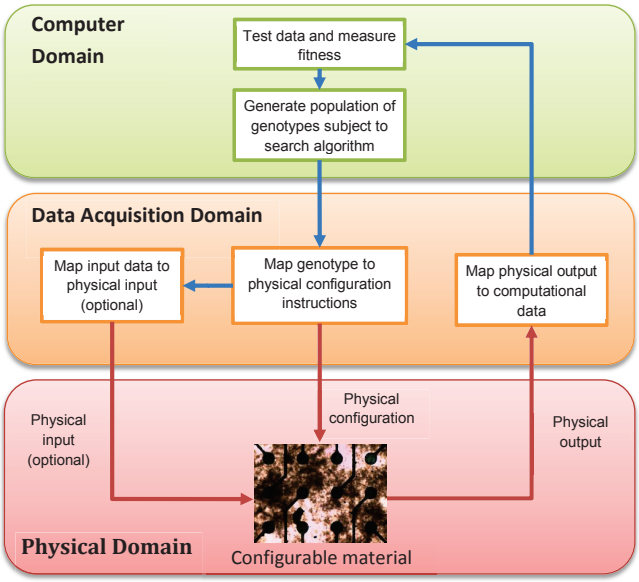

Fig. 3.1 Overview of *in materio* computing from (Clegg et al., 2014a)

2014b). This is appropriate when the system is not fully understood, as the fitness of a configuration can be calculated without necessarily having knowledge of how the computation is performed. Indeed, this is often the great strength of the approach, as it raises the possibility of exploiting hitherto unknown properties of the computational substrate.

> "The concept of [*in materio* computing] grew out of work that arose in a sub-field of evolutionary computation called evolvable hardware [[30, 38, 76, 104]] particularly through the work of Adrian Thompson [[86, 91]]. In 1996, Thompson famously demonstrated that unconstrained evolution on a silicon chip called a Field Programmable Gate Array (FPGA) could utilize the physical properties of the chip to solve computational problems [[85]]." —(Miller et al., 2014)

An overview of the processes involved are shown in figure 3.1. There are three domains: the computer domain, which manages the evolutionary search; the data acquisition domain, which manages the i/o between the configurable material and the computer; and the physical domain, which generates the solution. The search commences in the lower box in the 'computer domain', by generating a set of genotypes to be evaluated. Each genotype specifies both the physical configuration of the substrate, and the input data for the computation that is presented to it. This gives maximum flexibility for searching the space of possible computations that can be performed on the input data. For each genotype, the material is reconfigured and the inputs are then processed and fed into the physical domain in which the substrate resides. The

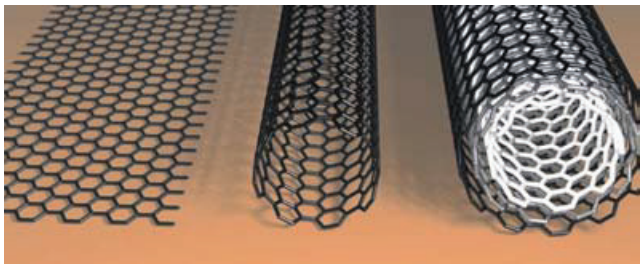

Fig. 3.2 Carbon nanotubes constructed from graphene sheets (Graham et al., 2005)

substrate is configured and processes the input data, and then produces outputs. The outputs are processed in the data acquisition domain and then used in a fitness evaluation of the genotype that specified the configuration.

The striking feature of this process is that there is a minimal requirement to understand the means by which the configurable material computes on the input data. Indeed, all that one needs to know about the material doing the computation that it is possible to configure it using a subset of the input channels. The inputs are usually electrical due to the fact that it is then straightforward to link the inputs to the genotype that is held on the classical computer, making the process much faster.

3.1.2 Carbon nanotubes

Carbon nanotubes are an integral part of the push towards further miniaturisation in classical semiconductor devices (ITRS, n.d.).

> "Carbon nanotubes (CNTs) were discovered by Iijima in 1991 while investigating the soot of an arc-discharge experiment used to create C 60 buckyballs [[2]]. From transmission electron microscope (TEM) images of periodic structures in the soot he speculated that concentric, graphene-based tubes had formed in the discharge zone. The basic forms of single-walled and multi-walled CNTs are shown in Fig. 3.2 together with a planar graphene sheet. One year later Hamada et al. suggested that these tubes could be metallic or semiconducting from tight binding calculations [[3]]. In the same year Ebbesen and coworkers presented an optimization of the arc-discharge method that yielded large quantities of CNTs [[4]]. Despite this initial success, it took a further four years to purify the arc-discharge material sufficiently to enable devices to be created, starting with the first measurement by Ebbesen et al. on the resistance of multi-walled carbon nanotubes [[5]]. Two years later the first transistor was made [[6]] and ballistic transport in multi-walled CNTs was established [[7]]." —(Graham et al., 2005)

> "The creation of a complete, functioning microelectronic system using only self-assembly and other nanotechnology methods is an enormous task requiring many

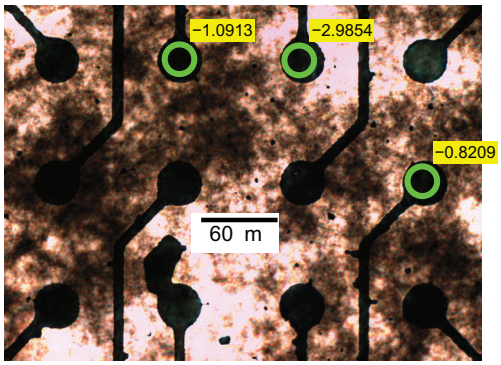

Fig. 3.3 Configurable carbon nanotube substrate embedded next to copper electrodes. From (Clegg et al., 2014b)

> different components to work together. Several important building blocks in this process have already been demonstrated, for instance selective bonding using DNA [[8]], but the ability to produce complex structures reliably seems a long way off. In particular, for microelectronics, it is important to reproduce the same device millions of times over almost flawlessly. For this reason a number of groups ... are focusing on the selective integration of specific bottom up components into traditional top down microelectronics process flows to create hybrid systems with superior properties." —(Graham et al., 2005)

> "Using carbon nanotubes as macromolecular electrodes and aluminum oxide as interlayer, isolated, non-volatile, rewriteable memory cells with an active area of 36 nm2 have been achieved, requiring a switching power less than 100 nW, with estimated switching energies below 10 fJ per bit [[35]]." —(Graham et al., 2005)

A key area of research in this domain is the challenge of reliability of the devices that are produced. Note however that the precision of devices is only applicable in CCOMP applications, and the UCOMP approach of *in materio* computation described in the previous section may provide an alternative route to exploiting these devices.

In (Clegg et al., 2014b), Clegg and Miller used a carbon nanotube assembly to solve simple travelling-salesman problems (TSP):

> "Here one wishes to find the minimum length tour that visits a collection of cities. A solution can be represented by a permutation of the cities. Such a problem is easily mapped onto an electrode array. In this case there are no input electrodes, only output and configuration electrodes. One allocates as many output electrodes as there are cities. The objective is to obtain a set of output voltages which can be mapped into a permutation of cities." —(Clegg et al., 2014b).

In short, each electrode represents a city in the TSP, and the ranked set of voltages at each electrode describes the best tour that the salesman should take. Initially, the voltage-ranking is random, but through evolution, the

voltages at the configuration electrodes affect the voltages at the output electrodes such that the correct solution emerges.

This methodology highlights some of the issues with this approach:

- It may be difficult and time-consuming to configure the device. The key design challenge of this approach is to find a convenient way of influencing the way the device processes the inputs.
- The device may not operate independently of its environment
- The substrate may have unwanted 'memory' of earlier events, which could influence the output of the current computation.
- Each implementation is a "one shot" solution: an evolved configuration on one device is not guaranteed to work on another.
- the result is merely a look-up table (although more sophisticated computations are in the pipeline).

However, the potential advantages should not be ignored (Miller et al., 2014):

- The response of the physical system does not have to be understood in depth.
- Unknown physical effects can be exploited, which may lead to new devices
- A solution can be found without any programming or design constraints, or limitations on human imagination and knowledge about the substrate.
- The substrate's computational capacity can potentially be exploited to the full.
- The amount of computation a small piece of material can do may be enormous.

3.2 Contribution to UCOMP Aspects

3.2.1 Speed

In the TSP application (Clegg et al., 2014b), *measuring* the chip's output may be relatively slow, but the actual processing rate of a trained chip is nanoseconds, due to the massive parallelism inherent in the substrate. This means that the approach would be good for complex recognition tasks as part of a robot architecture for example.

The speed of configuring the device may seem slow, since one has wait until the evolutionary algorithm converges to a solution. However, in a strict sense, this comparison should be made with the time it takes a human programmer to write a program to run on a classical computer (repeatability notwithstanding).

3.2.2 Resources

Although the CN chip is a low-voltage device, further research is needed to discover the limit on how low a voltage would be needed to make the system operate reliably.

Embedding CNs in liquid crystal is an approach that might yield further efficiencies in this area. The nanotubes move in the liquid crystal according to the voltages on the configuration electrodes. If the configured arrangement could subsequently be 'fixed', then the circuit would work with virtually no power at all. This approach also has the potential to make the CN assemblage easier to program for more complex tasks.

3.2.3 Quality

Considered independently of the programming criteria, these devices are potentially unbounded in their ability to carry out non-conventional computation.

> "A strong theme in evolution-in-materio is the evolution of novel forms of computation. It appears to be a methodology that has promise in building new kinds of analogue computing devices. Evolution is used as the programming methodology. If unknown physical properties and interactions are to be utilized in a computer program, it is obvious that the the program cannot be designed in advance. An evolutionary algorithm together with a fitness function rewards those configurations of the system that provide the desired computational response. To some extent, we can see the program itself as 'emergent', since it is not known in advance and emerges during the evolutionary process."(Miller et al., 2014)

3.2.4 Embeddedness

Although the computational function of the substrate is non-classical, we can not currently embed the system in a native environment due to the requirement for electrical configuration signals to "program" the device. If it were possible to "fix" the program, such is in the liquid crystal approach discussed in the Resources aspect above, it may become possible to embed the computational material in a variety of devices that use carbon nanotubes.

3.2.5 Formalism

There is at present no formalism for the *in materio* approach other than the use of search algorithms for programming. The best match is Shannon's work

on analogue computing. However, some of the pioneering work in the 50s etc. is not written up well. A lot of 'human in the loop' practises went on whilst training the systems. (See also work by John Mills/Lee Reubel in Indiana.)

3.2.6 Programming

Any stochastic search algorithm can be used to train in materio. This is because we don't know what the computational elements are. The process is analogous to training a neural network, with lots of back-propagation.

In terms of scalability, it may be possible to connect a number of slides together in a neural network like arrangement, using each CN chip as a hardware component. By this method, it may be possible to get some degree of programmability at a higher level.

> "We tentatively suggest that it will not be possible to construct computational systems of equivalent power to living organisms using conventional methods that employ purely symbolic methods. We argue that it will be advantageous for complex software systems of the future to utilize physical effects and use some form of search process akin to natural evolution." —(Miller et al., 2014)

> "According to Conrad this process leads us to pay 'The Price of Programmability' [[21]], whereby in conventional programming and design we proceed by excluding many of the processes that may lead to us solving the problem at hand. Indeed as Zauner notes, programmability is not a strict requirement for information processing systems since the enviable computing capabilities of cells and organisms are not implemented by programs. Artificial neural networks are also not programmable in the normal sense [[103]]." —(Miller et al., 2014)

3.2.7 Applications

In addition to the travelling salesman problem outlined above, *In materio* computation with carbon nanotubes has been used to develop solutions to function optimisation (Mohid et al., 2014b), machine learning (Mohid et al., 2014d), bin packing (Mohid et al., 2014c) and frequency classification (Mohid et al., 2014a).

At present it remains unclear what the best potential applications of this technology could be. It is anticipated that this should be revisited when some of the recommendations are implemented and challenges overcome.

3.2.8 Philosophy

Conrad criticizes traditional models of computing and he identifies that they have a common feature: at most very limited context dependency (Conrad, 1999). The logic gates in conventional machines implement formally defined input-output behaviours that are independent of the overall design of the circuit. Turing machines read and write symbols on individual tape squares in a manner that does not depend on the arrangement of symbols on the tape. Whereas computation in physical systems (particularly biological) are almost always context dependent.

3.3 Main Achievements so far

As detailed above, the potential for carbon nanotubes as a computational substrate is promising, but constrained by the difficulties in producing complex structures reliably. By using *in materio* computation methods, researchers are able to use the material in configurations that can currently be manufactured relatively cheaply. In summary:

- The core methodology has been proven
- Working hardware has been designed and engineered
- 'Software' (Implementation of TSP) has been tested and proven

3.4 What could be achieved in ten years?

Research in this area is focussed on demonstrating that devices can be programmed to solve complex problems faster – it is thought that this is reasonably feasible to do.

One way to prove this is to solve well known NP-hard problems. These solutions would act as a benchmark for the process and allow comparison with conventional CCOMP solutions. However, the focus on established CCOMP test problems merely explores solutions to problems that are well-known to be solvable by CCOMP devices. This is not necessarily the best application for CN chips, but perhaps gives the clearest demonstration of their potential.

An alternative goal would be to do the same thing as Shaw's factorisation of numbers on a quantum computer. The same potential exists here, but is achieved via exploiting the massive parallelism of the CN chip. It must be borne in mind that the 'true' application may be elsewhere – biological emulation perhaps (due to similarities in the parallelism between biological systems and the methodolgy described here.).

There is also potential to apply the techniques in other areas. For example the in materio evolution principle could be used to evolve bacterial colonies sitting in meshed combs of electrodes. Thus we could develop interfacing methods in biology.

3.5 Current challenges

3.5.1 Substrate

It has been difficult to get a large number of electrodes in one nanotube array. The Norwegian team in the NASCENCE consortium are working on this. it might be possible to buy electrode arrays rather than manufacture them, but it is unclear what the best method is for putting nanotube material onto the chip. A significant investment needed to achieve this, but this is unlikely to be secured without a demonstration of the potential first. Unfortunately, it might not be possible to demonstrate the potential without a large electrode array. A method needs to be found to balance this risk.

Another problem is the sheer range of potential *in materio* devices: what materials should we try? What concentration? What insulator? Without knowledge of the properties and with a richer model of *in materio* computation, we are conducting a blind search of the material space.

> "Harding et al. found that it was relatively easy to evolve a tone discriminator in liquid crystal – much easier than Thompson found it to be in silicon. Conceptually, we can understand this since most physical systems are likely to be endowed with many exploitable properties. We refer to this as *richness*. Silicon based electronic circuits have had the physics constrained as much as possible to implement Boolean logic and thus allow Turing computation. One way of possibly testing the relationship with richness and evolvability would be to attempt to evolve solutions to computational problems on a variety of FPGAs (say) with varying degrees of damage (perhaps by irradiating them). Experiments with radiation damaged FPTAs showed that evolutionary algorithms could make use of damaged parts of the chip... It remains to be seen if such damage actually helps solve certain computational problems" —(Miller et al., 2014)

Maybe we should try to understand the simplest possible implementation of an *in materio* system, and use this to guide the choice of further systems that use those principles. There remains the possibility that *in materio* computing might 'invent' a new technology by guiding a search of the possible physical arrangement of substrate.

3.5.2 Environmental effects, and the coherence problem

> "Ideally, the material would be able to be reset before the application of new configuration instructions. This is important as without the ability to reset the material it may retain a memory from past configurations, this could lead to the same configuration having different fitness values depending on the history of interactions with the material. " —(Miller et al., 2014)

3.6 Outlook and recommendations

> "One of the difficulties of analogue computers was that they had to be setup by experts who could create analogues of specific systems in convenient physical devices. Using a search process running on a modern digital computer gives us a new way to "program" a physical device. It also gives us a methodology for the discovery of new computational devices that use physical effects for computation in ways that human ingenuity has thus far failed to invent. " —(Miller et al., 2014)

References

Clegg, K. D., J. F. Miller, K. M. Massey, and M. C. Petty (2014a). "Practical issues for configuring carbon nanotube composite materials for computation". In: *ICES 2014, International Conference on Evolvable Systems: From Biology to Hardware*. IEEE, pp. 61–68.

Clegg, K. D., J. F. Miller, K. Massey, and M. Petty (2014b). "Travelling Salesman Problem Solved *'in materio'* by Evolved Carbon Nanotube Device". In: *PPSN XIII, Parallel Problem Solving from Nature*. Vol. 8672. LNCS. Springer, pp. 692–701.

Conrad, M. (1999). "Molecular and evolutionary computation: the tug of war between context freedom and context sensitivity." In: *Biosystems* 52.1-3, pp. 99–110.

Graham, A.P., G.S. Duesberg, W. Hoenlein, F. Kreupl, M. Liebau, R. Martin, B. Rajasekharan, W. Pamler, R. Seidel, W. Steinhoegl, and E. Unger (2005). "How do carbon nanotubes fit into the semiconductor roadmap?" In: *Applied Physics A* 80 (6), pp. 1141–1151.

ITRS (n.d.). *International technology roadmap for semiconductors, 2013 edition, emerging research devices*. www.itrs.net/reports.html.

Landauer, R. (1996). "The physical nature of information". In: *Phys. Lett. A* 217, pp. 188–193.

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

Mohid, M., J. F. Miller, S. L. Harding, G. Tufte, O. R. Lykkebø, K. M. Massey, and M. C. Petty (2014a). "Evolution-In-Materio: A Frequency Classifier Using Materials". In: *Proceedings of the 2014 IEEE International Conference on Evolvable Systems (ICES): From Biology to Hardware*. IEEE Press, pp. 46–53.

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

Mohid, M., J. F. Miller, S.L Harding, G. Tufte, O. R. Lykkebø, K. M. Massey, and M. C. Petty (2014c). "Evolution-In-Materio: Solving Bin Packing Problems Using Materials". In: *Proceedings of the 2014 IEEE International Conference on Evolvable Systems (ICES): From Biology to Hardware*. IEEE Press, pp. 38–45.

Mohid, Maktuba, Julian F. Miller, Simon L. Harding, Gunnar Tufte, Odd Rune Lykkebø, Mark K. Massey, and Michael C. Petty (2014d). "Evolution-In-Materio: Solving Machine Learning Classification Problems Using Materials". In: *PPSN XIII, Parallel Problem Solving from Nature*. Vol. 8672. LNCS. Springer, pp. 721–730.