Physical computation and embodied artificial intelligence

This thesis pertains to design engineering, electronic engineering, artificial intelligence, and computer science.

Offered for the degree of Doctor of Philosophy at the Open University.

Accepted 4 January 2005
Abstract

Physical computation and embodied artificial intelligence

In this thesis, we show that the Turing machine is insufficient to model the richness of physical behaviour. Instead, we propose a model of physical computation that allows us to understand how embodied intelligent agents can simultaneously be considered to be objects operating under the laws of nature (or physics) and information-processing devices. This analysis allows us to make concrete those issues relating to the relative merits of using analogue values or symbolic representations such as numbers. Indeed, our approach allows this longstanding point of contention in artificial intelligence to be transformed from a question of philosophy to one of physics. For well-defined tasks, the results of this analysis are equivalent to Shannon’s information theory. This is as expected. However, in certain circumstances, both the computational power and energy efficiency of an analogue system may be greater than would be possible using a classical Turing machine. This supports results from both neuromorphic engineering and theoretical computer science. Finally, for large, multi-functional systems with ill-defined roles, the new model provides a novel way of thinking about the complementarity of hardware and software.
I dedicate this thesis to two men who inspired me to make a contribution to science and technology: my father, Hardial Singh Bains, who died as I began my Ph.D.; and my mentor, Professor Stephen A. Benton, who died as I was finishing it. Thanks for challenging me to do and be more.
Acknowledgements


People who encouraged: Stuart Arnott, Lynn Bains, Rick Boardman, Fernando de Buarque Neto, Yiannis Demiris, Matthew Johnson, Kaveh Kamyab, Julie McCann, Bruce McLennan, Patrick Purcell, Sunil Rao, Alfons Salden, Arash Shaeri, Sydney Stanton, Angie Swain, and Mark Witkowski.

People who read: Paul Garthwaite, Steve Hart, Varol Kaptan, Silvio Macedo, Brita Olson, and Ian Pawson.

And finally, I’d like to thank my supervisor, Professor Jeffrey Johnson, without whom this study would not have been possible. His skill in identifying the strong and weak facets of an argument has been invaluable, and I am grateful to him for bringing out the hidden mathematician lurking inside me.
Contents

Figures: x

Chapter 1: Introduction 1

Chapter 2: Embodied AI as a practical engineering problem 7

Chapter 3: Artificial intelligence and computational power 35

Chapter 4: Analysis and model of physical computation 57

Chapter 5: Computing machines and noise 81

Chapter 6: Consequences of the model 95

Chapter 7: The interface with physics 121

Chapter 8: Discussion of the model's implications and use 143

Chapter 9: Conclusions and future work 167

References: 173

Appendix 1: Glossary of symbols 193

Appendix 2: Int'l Conf. on Systems, Man and Cybernetics 201

Physical computation and the design of anticipatory systems, 2004 (invited)

Appendix 3: Brain Inspired Cognitive Systems (conference) 211

Extending neuromorphic engineering beyond electronics, 2004

Appendix 4: AISB Journal 221

Intelligence as physical computation, 2003

Appendix 5: Joint Conference on Information Systems 239

Noise, physics, and non-Turing computation, 2000
Figures

**Figure 3.1:** The Turing machine.  
**Figure 3.2:** Venn diagram of possible computational-power scenarios in the current literature.

**Figure 4.1:** The intelligence function.  
**Figure 4.2:** The environment function.  
**Figure 4.3:** The system learning function.  
**Figure 4.4:** The environment learning function.  
**Figure 4.5:** The evolution of system and environment.  
**Figure 4.6:** The virtual machine and filter functions.  
**Figure 4.7:** The virtual machine learning function.  
**Figure 4.8:** Real interaction, where the filter functions are identity functions.

**Figure 4.9:** The virtual interaction case.  
**Figure 4.10:** Virtual interaction and the complementary function.

**Figure 4.11:** The virtual machine and filter functions.  
**Figure 4.12:** The virtual machine learning function.

**Figure 5.1:** Shannon's communication theory.  
**Figure 5.2:** Timing jitter as noise.  
**Figure 5.3:** Sample versus population mean for a time-encoded message.  
**Figure 5.4:** The width of the confidence interval.  
**Figure 5.5:** Indistinguishable values.

**Figure 6.1:** Physical information loss at an interface.

**Table 6.1:** Information retention for machines implemented on physical computers in different environmental scenarios.
Chapter 1: Introduction

This work considers some of the advantages of using analogue hardware in embodied, artificially-intelligent agents: particularly the computational advantages. To do this, we treat an analogue processor as distinct computationally from the Turing machine, with its own particular properties, and show that such a treatment supports analysis of the analogue machine as a super-Turing computer (one can perform all Turing functions and some others) in some circumstances. Using such a model not only has explanatory value—showing how Turing machines can exist within physical computing devices—but also have practical value for understanding how machines might best adapt and/or evolve to survive in their environments.

This work was inspired by neuromorphic engineering—where analogue circuits are favoured both for their similarity to biological systems and for their efficiency—and work that suggests that recurrent analogue neural networks are super-Turing in their computational abilities. These two have not, so far, been unified in the sense that the neuromorphic engineers make no claims that their machines are super-Turing, and the super-Turing theorists make no claims that their machines are physically realizable.

To find out whether there is a genuine correspondence between the two fields, a mathematical model is developed that builds a bridge between these
physical and computational regimes. This is applied to the particular application of a machine that survives and changes through its interaction—intelligent or not—with the environment. The model of physical computation shows the relationship between the agent, its environment, and the physics that couples them together. It also shows how the type of machine and the realities of physics affect the limits of the computational power of the agent. This has been much debated but, because most people make assumptions about the nature of physics (which is, in some basic senses, unknown), this has served to confuse the issue rather than to clarify it. In order to avoid this problem, we consider all the various physical realities that could exist, and explain why seemingly straightforward issues—such as the continuity, or not, of space-time and the meaning of noise in physics—cannot be taken for granted.

1.1 Objectives

In this thesis we show the following:

1. Why intelligent devices cannot be fully implemented through traditional computer science: that is, using the Turing machine model.

2. How both physical computation and Turing computation can be performed, simultaneously, by the same machine.

3. How the Turing machine, theoretically, can act as a constraint in modelling the power of the physical machine.
4. How our theoretical model can help us to examine the significance of unresolved physical questions and their impact on whether real or not machines can perform super-Turing computations.

5. In what contexts this new physical-computation model can offer added insight over traditional models relating to the design of computational systems.

1.2 Methods and structure
We achieve our objectives by first analysing the problem of artificial intelligence in a way that is conceptually consistent with both physics and computer science, and then building a model that reflects this analysis. The mathematical consequences of the model are then assessed for various possible physical scenarios. Next, we reduce the number of possibilities by eliminating those that do not conform to any of the physical scenarios supported by orthodox physical theory or current avenues of research. Only those consistent with physics, by this definition, are considered applicable to engineering. Finally, we look at these theoretical machines in conjunction with the other advantages and disadvantages of the physical computational approach to artificial intelligence.

In Chapters 2 and 3 we start by reviewing the literature from the perspective of application and theory: artificial intelligence hardware, and potential computational power, respectively. Chapter 2 concentrates on the practical: approaches to building intelligent machines, both analogue and
digital, and the stringent requirements that must be met in terms of power (both computational and energetic), speed, and size. Chapter 3, conversely, looks at more abstract issues: specifically, the theoretical capabilities of natural and artificial agents and how they differ computationally.

What emerges is a gap. On one side is engineering work that shows that analogue electronics are more efficient for neural-network-type intelligence tasks than conventional digital machines, and on the other is theoretical work that suggests that certain analogue neural networks may be computationally more powerful (can perform functions otherwise unavailable to conventional machines) than their digital counterparts.

In Chapter 4 we analyse, in simple physical terms, the mechanisms through which objects can sense, actuate, and learn. This analysis is incorporated into our model of physical computation that relates the nature of the interaction between an agent and its environment with its computational power, thus bringing practical and theoretical issues together. To allow an interface between the physical and the abstract (hardware and software) we also introduce a concept whereby the interpretation of a subset of physical inputs and outputs can be considered to implement a classical computation. Note that this is a general model that applies equally to intelligent agents, water droplets, or planets.

In Chapter 5, we define various types of computing machine—anologue, digital, and Turing—and the sets they operate on. We pay particular attention to the addition of noise to the analogue machine. Then, in Chapter 6, we plug
these definitions into the model and consider how the machines operate given certain assumptions about physical reality. Here, physical information retention provides a metric with which to test the machines as implemented in different physical environments.

For these results to have meaning, they must relate to the real physical world. In Chapter 7 we provide an interface between the options in our model of physical computation and the nature of reality. It is beyond the scope of this thesis to resolve these questions as they are a matter of research, and often controversy, within the physics community. Instead, we look briefly at the issues that are important to the model, why they are as yet undecided, and how different outcomes in physics would produce different results for us. This can also be considered in reverse: belief in a particular computational outcome might imply a belief in one physical interpretation of a theory rather than another.

To move back from the abstract to the practical, in Chapter 8 we consider the engineering implications of the current situation (where we have no conclusive answers to some relevant physical questions). We also consider the design implications of a combined physical/computational approach.

Finally, in Chapter 9 we conclude and consider further work.

**Conventions**

Throughout the thesis, terms to be defined that are specific to the model will appear in **heavy italic**. A complete list of model-related symbols and their
abbreviated definitions can also be found in Appendix 1 (the Symbol Glossary).
Chapter 2:

Embodied AI as a practical engineering problem

This chapter is concerned with the design of hardware that will implement embodied artificial intelligence, ideally to the point that a machine will be able to react to the environment in ways that make it appear to be as intelligent as a human. By starting with hardware, rather than cognitive models or connectionist networks, we do not intend to imply that software is unimportant. Rather, we want to show that—with a task as challenging as human-like AI—hardware may represent a bottleneck that we cannot afford to ignore.

2.1 Human-like embodied AI

To engineer a system, we must first devise a specification. Such a specification can be written at many different levels, and it makes sense for this application to start at the most abstract. Here, a human-like embodied AI is defined as follows:

S1. The machine should be similar in size to a human, say less than $2m^3$, be self-contained (except for some kind of fuel or other energy supply), and operate under a range of environmental conditions comparable to that of a human.

S2. The machine should be capable of sensing the environment and performing many physical tasks as well as humans can. Such tasks
would include playing games (such as soccer), cleaning, and interacting with people appropriately (including detection of the emotional states of humans). We would expect that there would be some tasks that humans could perform that, while impossible for some machines, would not preclude them from having human-like intelligence: for instance, a humanoid machine might be unable to swim because of its weight and density.

S3. Like a human, the machine should adapt to its environment, performing those tasks that are relevant to its interests, and which are within its physical ability. (Here, ‘interests’ could mean the machine prioritises its own survival, or that it works towards some end: like the positive outcome of a project or the good of society.)

We can break this down into the following elements: sensing the outside world; making decisions based on the information acquired; actuating to change the world as necessary; and learning from experience. If we do this then some potential problems become apparent. How do we know if the machine is sensing the world sufficiently in terms of number, type, and resolution of sensors? Does it have enough computing power to process this information appropriately and make decisions about what to do next? Can the actuators implement these decisions sufficiently well? Do they have high enough speed, accuracy, and degrees of freedom to make the appropriate action? Finally, is the latency of the entire system (from sensor to actuator) sufficiently low that circumstances will not have changed dramatically (at the
appropriate macroscopic physical level) by the time actuation takes place.

There are additional engineering constraints: power, size, and heat dissipation among them. We are unlikely to want to build an android that requires its own power station to operate. Or that is too heavy to walk on the floors of our homes and offices. Or that singes our hair if we get within ten feet.

This thesis, it should be noted, is restricted to embodied intelligences, not to those where access to resources is essentially unbounded. Examples of the latter include proposed distributed intelligences held on large networks of computers (e.g. (Maes, 1989)). This latter approach, which has its own advantages and disadvantages, is clearly not compatible with fully-autonomous, embodied AI. It may, in its own right, be a perfectly valid route to the first computer intelligence, but would not qualify as human-like in the sense that it is not embodied. Of course, a link to such an intelligent network would clearly be of use to the embodied machine (as it would to a human) in the right circumstances, but our robot must also be sufficiently autonomous to do without.

2.1.1 Dependence of hardware on application

From a hardware point of view we can see that there are sufficient issues to attend to without needing a full knowledge of the algorithms to be run. Further, hardware constraints might lead us to prefer some approaches to AI over others. The application constrains the hardware, and the hardware may
It is illuminating to look at an example of this process for another application. In the 1940s, when computer science was in its infancy, both analogue and digital computing were considered viable options. Scientists were interested in performing simple mathematical operations: the early computers were numerical calculators (e.g., (Ferry, 2003)). If the calculation was performed in an analogue system, two major problems could occur. First, the amplifier might not be manufactured perfectly to specification (typically ±1% would be considered reasonable even now, see e.g., (Fairchild Semiconductor, 2003)). So, if the original signal strength was assigned the value ‘1’ then the result of a multiplication by 5 could be 4.95 or 5.05 rather than 5 exactly. If this result were fed back into the next calculation, say another multiplication by 5 using the same circuit, the error could get even bigger. Eventually the result could be completely meaningless. Second, a perfect analogue amplifier that multiplies exactly by 5 could multiply the wrong thing. Noise could make the 1 look like 1.12 or 0.98. Again the result of the multiplication would be inaccurate.

These problems seem to disappear when using a system of digital logic. Whatever happens, the input and output in a digital system have to be either 1 or 0. If an incoming signal is 0.45 units it will still be treated like a binary digit (probably a 1, depending on the threshold chosen), and the answer will be a function of the implied 1 or 0, depending on the context. For a NOT gate, for instance, the answer would be 0. This makes digital logic very robust. Because
the answer to each logical 'question' can only take two values, it is impossible to be a little bit wrong. Hence, there is no small but growing error that can send the calculation off in the wrong direction.

What can happen is that the system gets the 1 or 0 value completely wrong. If the threshold for a particular device is set at 0.2, for instance, a very noisy (perhaps overheated) system might take it for a 1 instead of a 0. This produces a very large error, but computers are designed specifically to minimize the probability of this: designers work towards a low bit-error rate (number of errors per bit processed), ideally less than $10^{-12}$ (see e.g. Agilent Technologies, 2003)). Such unlikely errors are acceptable when doing arithmetic operations: especially when you have some means of checking that there has been an error (e.g. either by performing the operation more than once on the same hardware, or on two different circuits).

When digital technology neared a practicality threshold, analogue systems were abandoned for high-precision calculation, and a whole theoretical and technological structure was developed to support digital computers. As a result, today's problems are formulated to run on digital computers even when this is slower, consumes more power, and requires a physically-larger processor than an analogue machine performing the same calculation.

That said, because of their remarkable efficiency, slide rules remained tools of the engineer until well after the original pocket calculators were invented.
2.2.1 Neuromorphic engineering

Despite the well-known exponential rise in the power and use of digital computation (Moore, 1965), Carver Mead pioneered a new way of exploiting analogue electronics. His approach came from an interest in implementing neural circuits to process sensory information. In a book that has come to define the field of neuromorphic engineering (Mead, 1989), Mead showed that brains are about a billion times more power efficient than conventional computers. Though part of this could be attributed to the inefficiency of the individual transistors, etc., he believed that a performance improvement of six orders of magnitude was possible based on two things: using less wire and more local interactions; and using, not suppressing, the intrinsic physics (i.e. the analogue nature) of the device.

The first of these is an architectural strategy that is helpful for any designer, whether working in digital or analogue. In an electronic system, the length of wire between one component and another acts as a power drain. Component A only really wants to charge up component B: nothing more is necessary for the computation. But to do that, it has to charge up all of the wire between the two components also. The extra power required to do this fulfills no computational purpose: it is merely an architectural overhead. Though designers spend a lot of time trying to keep the amount of wire to a minimum, the kinds of processes that we run on digital computers (and the gate layouts that make them possible) do not lend themselves to the ideal architecture: one in which devices only have to communicate with near
neighbours. The opposite is true of most neural networks, which also have major advantages for sensor fusion (see e.g. (Klein, 1999)). By changing to this more efficient interconnection scheme and processing type, a 100-fold power efficiency gain can be made.

However, the advantage of moving from a digital to an analogue neural system is potentially even greater. Mead has shown that computation could be performed up to 10,000 times more efficiently (in terms of electrical power) by not throwing away the true functionality of the electronics used. In other words, by using them in a more analogue way. Consider, for instance, a memory cell. Intrinsically, there is nothing about such a circuit that forces it to be either ‘full’ or ‘empty’ (to contain logic 1 or logic 0): we simply choose to interpret the information that way when we read it out. Because of this choice, we are forced to use an extra cell for each bit of information we want to store. If we allowed the charge within the cell to vary continuously (or at least to be able to adopt different ‘grey levels’), then we could store significantly more information in a much smaller space. In addition, we might need to use less energy to both store and read out the information (because we would only have to go to one location to get it).

The same kind of analysis is true on the information-processing side. The electronic circuits we use in computing have interesting, and often useful, responses. For instance, they can perform multiplications or manipulate a signal differently based on strength or timing. Some of these features are, says
Mead, very similar to those we see in the brain's processing devices. But we do not really make use of these as computational primitives (low-level processes on which higher ones can be built). Instead, we reduce problems to AND, OR, NOT, etc., and force our hardware to give us the minimalist answers to these questions.

From a power-consumption point of view, it is important to note that the forcing required to achieve Boolean logic involves driving the electronics in a particularly inefficient mode (though this is being addressed to some extent by those working in reversible logic). More energy is consumed for an incoming signal to trigger a sharp non-linear (thresholded) response, than a sub-threshold (closer to linear) response.

In effect, we are reducing the number of operations that each component can perform to decrease the probability of error, but increasing the energy that each takes. Given the example in the last section, we know why this approach is taken. If we need a precise answer, we would not want to use an amplifier to multiply one number by another. Depending on the noise and the extent to which the device varies from its specifications, we would be likely to get a different answer every time. Thus we string together many gates, each consisting of many different components, to perform a digital multiplication. We get a precise answer, but at a high cost in terms of efficiency.

If we have an application where we don't need that kind of precision, then this approach makes no sense: in this case we are throwing away the
useful physics of the devices themselves, recreating the same functions using logic gates, and have nothing to show for it at the end. In such applications, analogue computers can be vastly more power efficient without sacrificing functionality.

An illustration of this is the cellular neural network (CNN) invented by Leon Chua (e.g. (Chua, 1998)), and now being used as an image-processing tool in the vision community. The CNN is a device that is digitally programmable but can perform complicated non-linear operations during the analogue transient, the time to go from one stable state to the next. It is an apt demonstration of Mead’s point. Not only is the device far lower in power consumption than the equivalent image processor, but can also be orders of magnitude faster (depending on the algorithms implemented).

2.2 The challenge of building brain-like systems

In principle, an artificially-intelligent agent (even if intended to act like a human or other animal) need not necessarily work in a biologically-inspired way. There are many conceivable implementations of sets of behaviours we would consider to be intelligent or even human-like. That said, the only existence proof we have of a physical machine that can implement human-like intelligence is the human. If we are copying human behaviour, studying—and even functionally replicating—the mechanisms that lead to this behaviour may make sense as well. The brain has on the order of $10^{12}$ neural processors, linked by as many as $10^{15}$ synapses (Churchland & Sejnowski, 1992), all
contained in an object about the size and topology of a folded pizza (analogy from Christof Koch, California Institute of Technology). If each of these parallel processors were performing just one logic operation per emission of a neural spike, duplicating this system in digital electronics would be formidable challenge. In fact, according to Yaser Abu-Mustafa, even this simplified artificial brain would be practically impossible today because of the lack of connectivity between artificial neurons (Abu-Mostafa, 1988a; Abu-Mustafa, 1988b). In his papers (and echoed in an appendix he wrote for Carver Mead's book (Abu-Mostafa, 1989)), Abu-Mostafa argues that the ability of a local-learning neural network (which would hold for the biological case) is limited by the number of connections between the neurons. This, he says, cannot be compensated for by simply using a larger network.

If biological neural networks do indeed fall into this class, then this could represent a major bottleneck. There are three possible options. One is to find some kind of learning rule that allows us to produce the same discrimination ability with a modest level of interconnectivity—on the order of 1-10 connections per neuron rather than the average of 1000 in biological systems. Another option is simply to find some way to improve connectivity. Finally, we could just give up on the neuromorphic approach entirely. It turns out that the first and third of these options may be the same. According to Abu-Mustafa, the local learning rule was considered precisely because it was biologically plausible: departing from it (necessary to overcome Abu-Mustafa's limit) by definition changes the nature of the system being designed.
Since dense connectivity is an issue that has had to be addressed for many connectionist (neural-network based) and distributed-computing applications, we will consider the second option—improving connectivity in artificial systems—here.

### 2.2.1 Connectivity as a hardware problem

David Miller (Miller, 1989, 1997; Miller & Özaktas, 1997) has written extensive critiques on the problems of on-chip electronic interconnections. The problem can be expressed simply: the longer an electronic interconnect is, the higher its capacitance. This means that more energy is required to get a signal from one end to the other, and, since circuits have to be designed with the worst case scenario in mind, the performance of the entire system ends up being determined by the longest link. As a result, designers try to keep interconnects as short as possible: nearest-neighbour interconnects are ideal. Similarly, the more interconnects a chip has (of whatever length) the more power it must produce to charge them all simultaneously. This is particularly true when broadcasting a signal: sending through all possible interconnects at once. Thus the number of interconnects becomes a limiting factor. Since broadcasting does seem to be an important part of some neural functioning (see e.g. (Abu-Mostafa, 1988b)), such scaling issues will be considered.

When information must be exchanged between one chip and another, there is a further problem: a bottleneck is caused by the fact that electronic die are essentially two-dimensional. If a square chip is length x on a side, then the
area (and number of processors accommodated) varies as \( x^2 \), while the number of interconnects (pins along the edges of the chips) varies only as \( 4x \). As silicon wafers get larger and feature sizes get smaller, this problem gets worse (even if the size of pins scales down too).

Next, there is the problem of crosstalk. Interconnects must be electrically shielded from each other, otherwise the electric field will create false signals (noise) in neighbouring wires. This leakage also causes power dissipation. The need for shielding constitutes a lower limit on the volume required for an electrical interconnect.

Finally, light has a speed advantage too: the time difference in arriving pulses caused by long and short light paths (whether free-space or guided) is negligible compared to the time between the pulses themselves. For this reason researchers at Cray have used fibres to reduce clock-timing jitter in high-performance machines.

Inventive ways have been found to get around these problems for experimental systems. The best known electronic method, known as address-event representation (AER, Boahen, 2000; Mahowald, 1992)) essentially gets around the broadcasting problem by using a time-multiplexed, pulsed network. In this asynchronous (analogue time) system, individual pixels request access to the bus when active. For example, Eugenio Culurciello (Culurciello et al. 2001) built a system where pixels in an imaging array, via artificial neurons, emitted spikes at a frequency proportional to the light intensity. Since AER is a responsive network—access to the communications bus is granted on
request—the available bandwidth is allocated according to need. Also, because the spike time is small in comparison with the inter-spike-interval time for a given neuron, delay in getting access to the bus does not significantly change the pulse-coded signal.

Unfortunately, though it represents an ingenious and efficient use of infrastructure, AER does not change the fact that there is a serious scaling problem. It was designed to scale well (which it does) for increasing numbers of processing elements on a single array: for \( N \) elements in an array to connect to the same number in another array, only \( 1 + \log_2 N \) wires are needed. If \( M \) such arrays have to be connected with each other, then \( M (1 + \log_2 N) \) wires would have to be in place. With current technology, only relatively small numbers of chips can be connected in this way. In addition, AER is not designed for on-chip communication, so there is a trade off between making the arrays bigger (for fewer chips in the network) and local connectivity.

There are other techniques under development, some of which are likely to be compatible with AER. Most of these would fall into the broad class of optical interconnects. Unlike wires, optical signals don't have crosstalk (unless handled poorly at the detector), and so can co-exist in the same guiding volume. This comes from the intrinsic physical fact that photons (as bosons) neither repel nor attract each other, nor do they interfere with each other in the way that electrons (fermions) do. (The wavefunctions of photons do
interfere in some conditions, but that is another issue, and not relevant here).

The most obvious example of this is the wavelength-division-multiplexed fibre-optical link (Miki & Ishio, 1978) where hundreds of closely-spaced wavelength channels can propagate down the same optical waveguide and be differentiated at the other end without their signals becoming mixed. As the light sources, non-linear materials, detectors, etc., for photonic systems become cheaper and more available, increasingly ambitious systems are being built. For example, Nan Jokerst built a substrate-guided or planar-optical system at Georgia Institute of Technology (Jokerst et al. 2000). Essentially, this gives the light freedom to move in $2\frac{1}{2}$ dimensions: the two dimensions of the plane of the interconnect, plus up and down (into devices attached to either side of the plane). By including emitters and detectors on the top of circuits, signals can be either actively or passively routed to other locations (potentially many other locations) on the chip or board.

Among many other notable examples is NEC's optical backplane network at its Laboratories in Princeton (Araki et al. 1996). This was aimed at providing slow reconfigurability for massively parallel computer systems and used free-space optics (photons in air/gas) rather than guided-wave optics. Based on an analysis of distributed computing systems, the NEC network was designed to use an electronic crossbar for local interconnections on each board, and had vertical-cavity surface-emitting lasers (VCSELs) emit signals to travel between boards. For the latter to happen, each needs a VCSEL and detector for every other board in the network, and the address is specified by
the geometrical location of the VCSEL in the array. This kind of scaling may seem similar to the one that posed such a problem in AER. However, optical paths can cross without affecting each other and, since they do not have such severe problems with heat dissipation, they can be arranged in close-packed arrays, with the area of the array scaling linearly with the total number of interconnected boards.

A more radical solution, using both fibres and free-space, was designed by Ed Frietman at the University of Delft (Frietman, 1995). Rather than producing a network that on some level requires message passing, like the NEC and AER schemes, this system was based on every processor talking to every other processor. This works by each sending out information through an array of light-emitting diodes, one for each bit to be transmitted. This light is then captured by a polymer fibre optic array and connected to a central node known as the kaleidoscope. The fibres are organized so that their relative positions at the output are the same as at the input: effectively making the data into a 2D image. In the kaleidoscope, the fibre bundles are tiled to make one large 2D image, the size of which is dependent on the number of bits output by each processor and the number of processors in the array. Using a faceted mirror and lens system, the light from the entire image is broadcast onto separate locations for each processor, coupled into a second fibre bundle, and then received so that the processor can access the data electronically.
Not all current strategies to increase connectivity are optical. Irvine Sensors researchers invented a mechanism designed to be compatible with its chip-stacking technology (Carson, 2000). The stacking technique involved producing essentially conventional chips and removing the substrate (wafer) through polishing. Apart from allowing more devices to fit into the same volume, this also gives a new opportunity for interconnection. To capitalize on this they invented the three-dimensional field-effect transistor. This device is constructed as two parts on two separate chips: once they have been bonded together in a chip stack, they operate as a single transistor. This means that the concept of a nearest neighbour is extended to three dimensions rather than two: as it is in the brain. This work continues.

Interestingly, this research is likely to be complementary to the optical approach rather than competing with it. Irvine Sensors has a history of supporting research into optical interconnects (see e.g., (Li et al. 2002)) and its scientists have admitted that—to build something the size of the brain—many stacked silicon modules would be necessary (personal communication). One suggestion is that these modules could eventually be connected using free-space optics, both to provide the appropriate interconnection density and to ease heat dissipation.

### 2.2.2 Capturing the complexity of neuron interactions

From our earlier discussion of Carver Mead's work, we know that the replication of simple analogue functions is more efficiently performed using
Physical computation and embodied artificial intelligence

sub-threshold electronics. However, there is much evidence that real neurons are more complicated than even Mead's circuits (see e.g. (Churchland & Sejnowski, 1992) or (Wehr & Laurent, 1996) for more specific examples). Analysis of one of the best-known neural circuits, the Hodgkin/Huxley model of the squid axon (Hodgkin & Huxley, 1952), suggests that it performs in a mathematically complex way (see e.g. (Arcas et al. 2000)). The excitable membrane of the squid's axon becomes increasingly depolarised due to the incoming signals (spikes or pulses of ions) until it reaches a threshold potential. At this point it abruptly inverts, producing the neuronal action potential and, thus, the output signal. Afterwards the membrane undergoes a period of no response, followed by another slow build-up. As a result, a pulse arriving immediately after firing will have a completely different effect on the output of the neuron than a pulse arriving immediately before. Thus, spike timing is critical.

Nabil Farhat has done significant work in this area, both from the dynamics (Farhat & del Moral Hernandez, 1996) and hardware perspectives (Farhat & Wen, 1995). He claims that the real complexity comes from combining this behaviour with the processing carried out by the neuron's receptors in the dendritic tree. Correlations, caused by synchronicity in the incoming signals, cause a periodic modulation at the excitable membrane. This gives rise to complex, ordered patterns of firing that are phase-locked to the periodic modulation, as well as disordered (chaotic) firing that depends on
the amplitude and frequency of the modulation. In this way the neuron detects coherence (meaning) in arriving spike trains and encodes this information in its own output.

An analogy can be made here between spike timing and optical phase. In photography, only the combined intensity of light arriving at the film is recorded: the phase information (encoding distance and direction) is lost, resulting in a flat image. In holographic imaging, the phases of incoming rays are allowed to interact through interference, and the result of that interaction is recorded (Hecht, 1987). This encodes the image's third dimension, depth, and increases the amount of information that can be stored and retrieved (Mok et al. 1992). In fact, researchers have exploited optical interference to make associative memories (e.g. (Farhat et al. 1985)), based on optical correlators (Casasent & Psaltis, 1976; Juday et al. 1987) combined with holographic data storage (Psaltis et al. 1990). Such devices have the advantage of both high density—through efficient use of material—and content addressibility: possible because the geometrical/topological integrity of images is retained in the way they are stored.

Farhat built simulations and actual analogue models of neurons (analogue hardware) that could take timing into account in this way and found that they produced a bifurcated output: they depended on only small changes in input frequency and phase to produce periodic, m-periodic, quasi-periodic, and chaotic firing patterns. Further, Farhat designed a system that could take advantage of the high-connectivity available through optical interconnects, the
complexity of analogue neurons, and the unusual properties of electron-trapping materials (ETMs): the latter added a further layer of (biologically-plausible) nonlinear interaction to the network.

It is worth examining how the additional complexity is achieved. ETMs are materials that contain two sets of impurities: one with an electron that is easily liberated (Eu$^{2+}$) and another that provides a trap for it (Sm$^{3+}$). On illumination with blue light, electrons are excited and either fall back to the Eu$^{2+}$, producing orange fluorescence, or become trapped. On illumination with infrared light, trapped electrons tunnel back to the Eu ions and then fall into its ground state, again producing orange light. Electron beams can be used instead of the blue light, increasing the complexity of the interaction. Normally these materials are used in a read-write series, where the IR reads out what the blue records. However, when IR and blue light illuminate the ETM simultaneously, the dynamics become complex: especially if one or both beams change with time. As a result, Farhat determined that the ETMs could provide input and output (dendritic and synaptic) weights, nonlinearly coupling bifurcation neurons together.

Farhat has shown (Farhat, 1998) that, in simulation at least, the bifurcation neurons formed nonlinearly interacting phase-locked netlets or neuronal assemblies with external input. The chaotic periods served as noise, and assisted neural functioning through a kind of stimulated annealing process: they helped the netlet to converge by popping it out of local minima.
According to Farhat, this behavior was very robust at the netlet level, even if the reactions of particular neurons were imprecise. Further, he says that this behaviour seems to mimic that of cortical neurons and will be potentially useful in creating intelligent machines. Experiments with similar networks (Farhat, 1997) have shown that they can be used to accurately discriminate between objects. Similar conclusions about noise have been reached by researchers at Boston University (Mar et al. 1999).

The fact that complex analogue neural behaviour exists in nature does not prove that it is functional, but Abu-Mostafa argues that it is. In his appendix to Mead's book (Abu-Mostafa, 1989), he shows that not only does using analogue neurons with more than two inputs reduce the number of neurons required (to replace $N K$-input analogue neurons, a minimum of $NK^2$ binary neurons would be required), but there is also no guarantee that the binary network would do the job.

That said, there is no reason in principle to argue that neuron complexity cannot be traded-off against speed, network size, and so forth. However, in practice (and this chapter is concerned with the engineering aspects of systems), though imperfect, biology is generally very efficient. For example, the use of ATP (adenosine triphosphate) as a fuel for motor proteins (such as kinesin and myosin) is highly efficient (Howard, 2001): it has almost no thermal by-product, turning chemical energy almost entirely into mechanical energy. By over-simplifying neural processes and ignoring the complexity of biological exemplars, we risk turning human-like embodied AI from a problem
that could be solved practically (implemented in a machine the size of a man) into one that cannot (where the machine would have to be the size of a planet).

2.3 Hybrid system design and A/D conversion

Most work in embodied artificial intelligence (see e.g. (Johnson & Picton, 1995)) has a conventional structure: sensors receive analogue information from the outside world, turn it into bits, and send it to be processed by a digital computer. Once the processor has decided what to do, the decision is passed to a controller that turns this information into the appropriate (often analogue) driving signal for the actuators. Such machines can be defined as hybrid—both analogue and digital—and their structure as conventional: the A/D and D/A conversion stages necessitate the isolation of each of the sensor, processor, and actuation stages from each other.

It is not always necessary to design systems in this way: instead, all three stages can be merged into a single physical system. Such an approach has two main advantages. The most obvious is in terms of speed and power consumption. This fact is increasingly being recognised by roboticists. For instance Matthew Williamson, who worked on Rodney Brooks' Cog project, was charged with engineering the robot's limbs in such a way as to ease both the information-processing and energy burden they represented for the machine as a whole. A mechanical engineer by training, he designed Cog's arms and wrists so that they were compliant (Williamson, 1995) and could
respond mechanically to changes in the environment rather than purely through conventional sensor-processor-actuator loops. The result was a considerably lower computational overhead and a more mechanically-efficient and natural-looking movement.

Others interested in exploiting a balance between information processing and a machine's physical dynamics (in this case, mechanical) include Lungarella and Berthouze (Lungarella & Berthouze, 2002). They have looked at how temporarily restricting the degrees of freedom of a mechanical system can improve a robot's ability to learn how to manipulate it. In another recent paper, Pfeiffer gives numerous examples of the importance of mechanical design to machine intelligence (Pfeifer, 2002). Going further back, biologically-inspired roboticists have looked at how cats survive falls from tall buildings through a mechanical process of turning, and through using their legs as a buffer between critical systems (the body and brain) and the ground (Cameron & Arkin, 1991).

Another advantage of avoiding the conventional setup is that it is not necessary to determine what resolution of A/D/A conversion is necessary. Conventionally—in the case of machine vision, for instance—an engineer has to consider how many grey levels are necessary to implement a particular task (such as floor-following (Horswill, 1993), defect detection (Davies, 1990), or number recognition (Hinton, 2000)) in a particular set of conditions (e.g. over a given range of light levels, object orientations, distances, etc.) in order to specify the hardware.
Though machine vision is still advancing, the reverse-engineering process for choosing sensor sensitivity, resolution, and dynamic range is generally well understood and is used for most applications. However, because one of the prerequisites for such reverse engineering—a clear specification—is unavailable for our application, the interface between analogue and digital layers becomes a problem. This fact is intrinsic to point S3 in our definition of human-like AI given earlier: the machine must be free to adapt to its environment based upon what's important to its survival. This suggests that the machine should be free to use the dynamic range of its sensors as it sees fit, without an artificial A/D conversion limit. If the maximum bit resolution is specified in advance, then information that came in through the analogue sensors could be withheld from the processor by the A/D conversion. With this information unavailable for scrutiny, the embodied agent would never be able to learn whether it was important or not.

In humans and other animals, sensors output their information as asynchronous spike trains: the timing between spikes can vary continuously and is not regulated by a clock. This freedom from being forced into discrete values can be useful. For some problems, non-linear greyscales (such as logarithmic, see Weber-Fechner law, e.g. (Walker, 1995)) are preferable to linear: particular light levels may more important and require more discrimination, others less. Further, if the scale can be changed for different applications or environmental conditions, then whatever information is available may be exploited to its maximum potential. Most human senses,
including touch, vision, and audition, have such non-linear sensitivities.

Perhaps the clearest example of where A/D/A conversion can be a problem is in systems where feedback is important. In electronics, positive and negative feedback are used to either minimize or maximize small changes in an input, thus making a circuit either stable to small changes or extremely sensitive to them (Young, 1988). A nice example can be drawn from optoelectronics (Dupertuis et al. 2000) where a semiconductor optical amplifier (SOA) can be made several times faster without reducing gain or increasing current through the injection of a continuous-wave light beam. The technique exploits the transparency point of a photonic device—where a wavelength is absorbed and emitted equally—using the incoming light to boost the production of carriers when they are most needed. It works by solving a specific problem: when the incoming signal at one wavelength is amplified by the SOA, there is a decrease in the number of charge carriers available to produce gain and, electronically, it takes a long time to replenish. However, the decrease in charge carriers has a secondary effect: it shifts the transparency point of the material. The continuous-wave beam, which is at a different wavelength, is at this transparency point, having almost no effect on the device before the signal arrives. While the signal is present and amplification takes place, however, the resulting drop in charge carriers pushes the injection wavelength into the device’s absorbing region, and the absorption, in turn, quickly produces the needed charge carriers. Thus, the device implements an optoelectronic feedback loop.
Another example, particularly relevant for our application, is local inhibition in the retina (for a brief review and recent developments see (Roska et al. 2000). Here, small differences in the intensity of light received are amplified so that only the brightest pixel amongst nearest neighbors fires. This principle has been incorporated into neuromorphic systems called winner-take-all networks (Lazzaro et al. 1989) and these have been used in navigation and sensing (e.g. (Indiveri et al. 1996)). Part of their advantage comes from their analogue nature, which means it is almost impossible that two incoming signals can appear to have the same intensity. Even the smallest difference can be leveraged to produce a clear winner. If digitized, this would not be the case.

2.3.1 Fusing interdependent sensory information

Another area in which resolution issues can become important is sensor fusion: this is especially because of the inter-dependence of sensor systems within a complex embodied intelligence such as a human. In essence, the sensors perform an application that is more than implementing a particular sense. Specifically, the eyes must not only supply sufficient information to allow, for instance, visual pattern recognition to take place, but they must also supply sufficient information to guide and supplement (for example) locomotion or audition. Further, they must supply this information in an appropriate form.

To give a more specific example, in the spinal cords of many vertebrates
(Cohen et al. 1992), sensory information is directly incorporated into the gait of the animal through a process of entrainment (synchronization). Cohen and colleagues have modelled the interaction with visual feedback both theoretically and experimentally. The spinal cord has its own driving frequency, the source of which has been modelled as a coupling between adjacent oscillator sections (such as the vertebrae). The coupling takes place through the neural integration of spike trains, and so can be manipulated by the addition of extra spikes.

The word 'entrainment' refers to the modulation of one oscillation by another (similar) signal, with the output both frequency- and phase-locked to the modulator. This kind of entrainment does occur with some sensory input, particularly tactile sensors directly connected to the locomoting limbs. However, in the visual case, the signals are not of similar frequency: instead the integrating neuron reaches its threshold condition earlier when additional excitatory spikes are added, more slowly when the extra signal is inhibitory. In this way, the analogue signals from one sensory modality can directly interact with the behaviour of another system without any explicit processing.

That raw information may be shared among senses is a crucial point. Studies of the brain in many different animals (see examples for human, monkey, and cat in (Foxe et al. 2000), (Duhamel et al. 1998), (Wallace & Stein, 1997), respectively) have shown that visual and auditory signals are not just processed by their own processor (or cortex): there are connections that leave each of these sensory systems very early in the chain of processing.
Thus, the resolution required for visual applications may not be the same as for those systems that exploit visual information for non-visual tasks. As in the spinal cord example, such systems are not necessarily making use of the results of conventional visual processes such as pattern recognition, object tracking, and so forth. Rather, they are processing the visual signals in their own ways for their own ends.

Because the sharing systems—those directly exploiting sensory signals that are nevertheless outside their primary modality—are working with relatively unprocessed signals, they can potentially make use of any information that such signals contain. If the signal is digitised then the information available has been artificially restricted. Once again we return to the problem that, in order to be able to set a resolution that is indistinguishable from the original analogue, it is necessary to fully understand both the neural processes and the applications they are serving. Such an expectation would not be permitted from our definition of an embodied, human-like artificial intelligence given previously.

**Conclusions from this chapter**

- Highly-interconnected electronic (and opto-electronic) neural networks using a local learning rule scale better than binary-connected processors for the discrimination of environmental features.
- For applications with high computational complexity, analogue networks can be both smaller and more efficient than their digital
counterparts.

- Systems that use digital processing require A/D conversion, the correct level of which may not be straightforward to reverse-engineer.

- Where sensing, processing, and actuation are done by a single dynamical system, speed, energy efficiency, and the degree of sensor fusion may be increased.

**Assumptions and discussions beyond the scope of this thesis**

- Throughout, this work will take a materialist stance: that is, it will be assumed that all properties of intelligence come from the physical world. What is often described as mind-body duality will not be considered.

- Given the ill-defined nature of intelligence as a problem, it will be assumed that a) the functions performed by the brain may be mathematically complicated (e.g. chaotic), and b) many of the functions that the brain performs are unknown.
Chapter 3:

Artificial intelligence and computational power

In this chapter we review the literature pertaining to the computational capacity of machines, starting with the Turing machine, and consider the relevance of these models for the embodied artificial intelligence task.

3.1 The Turing machine as a standard of computational power

A Turing machine is a thought-experiment computer that defines the limits of what is possible using discrete-state machines (Turing, 1936), by setting up the limiting case of a machine with almost infinite resources. If a computation cannot be performed on a Turing machine, it is impossible to implement on any real digital computer.

The machine consists of two main components (see Figure 3.1). The first is a piece of computer tape that is either infinitely or arbitrarily long (Turing is not clear on this point). It is divided into individual cells that contain 'data' in the form of symbols. These symbols come from a finite, pre-defined set. The second component is a mechanism that reads one cell of the tape at a time and can both mark the tape and move it backwards and forwards. How the tape is moved or marked is dependent on a finite set of rules. Which rule is carried out is determined by the data and by the machine's 'state', a value that is held in memory.
The Turing machine is described as ‘universal’ because it was proven to be equivalent to a number of other theoretical models of computing (Turing, 1936). It has also been hypothesized that the set of functions that can be performed by algorithms—self-referential specifications of steps that produce the required answer when carried out—is the same as the set of functions that have been performed by these equivalent computing models (Church, 1936). Turing universality and the Church-Turing thesis are, perhaps, not surprising results because the models that they show to be equivalent are based on the same mathematical assumptions: they use finite sets of symbols and involve rule-based manipulations with similar degrees of freedom.

An advantage of the Turing machine model is that it allows us to understand what is and is not possible using digital computers. Understanding these limitations enables us to construct problems in ways that are
computable: computable by a Turing machine. For instance, Turing himself (Turing, 1936) pointed out 'the halting problem': where one program or algorithm cannot tell whether another program or algorithm will stop. In such cases, the only way to tell whether a program will stop or not in a given time is to run it and see, and the testing program can only report whether the tested program has stopped or not. This problem is rooted in the incompleteness theorem. Gödel showed that in formal mathematical systems rich enough to include arithmetic (including the Turing machine) suffer from a problem known as incompleteness (Gödel, 1931). This means that there are propositions that can neither be proved true nor false within the system: these are said to be undecidable.

In terms of artificial intelligence, Penrose (Penrose, 1989) likens this inability to prove certain types of statement to a lack of insight. He points out that human beings are able to tell certain statements are indeed paradoxes (such as the statement, "I am lying"). This ability, he says, involves stepping outside the logical rules to see the truth. Because of this, he claims that there are certain truths particularly mathematical ones that humans can see (he cites himself as evidence) and that machine algorithms, constrained as they are by the incompleteness theorem, cannot. Thus, he says, human-like artificial intelligence is impossible with machines that are constrained in this way.

It should be noted that there is an approach that gets around the halting problem, sometimes called computation in the limit (see e.g. (Kugel, 2003)).
Using this approach, the solution is to have the program that tests for halting immediately give the answer no: and then change to yes only if the program stops. This approach, described by Kugel as finding solutions by ‘trial and error’, has the advantage that it does not require infinite time. However, from a practical computing standpoint it adds very little: certainty in output (you always get an immediate and clear answer), is traded against the fact that the longer you wait, the more accurate the answer is likely to be: take the answer too early, and it is very likely to be not just slightly, but entirely, wrong.

3.1.1 Do these constraints matter in embodied AI?

That computers are constrained by Gödel's incompleteness theorem is not controversial in the literature. However, the implications of this fact are. For instance, Hofstadter (Hofstadter, 1979) raised and elucidated these issues before Penrose did, and in a clearer way, but stopped short of claiming that AI was impossible as a result. In fact, though much of his book could be seen as supporting Penrose, the chapters on complexity (ant colonies) in the middle of the book suggest that even simple algorithms can provide rich behaviours.

Another school of thought (LaForte et al. 1998), says that Penrose is wrong on the basis that he is assuming a kind of self-knowledge that humans don't really possess: even in the confined sphere of mathematics. LaForte and his colleagues suggest that human mathematicians are likely to have have the same underlying weaknesses. Regardless of whether or not they believe their methods to be correct, mathematicians cannot know that for sure.
Wegner (Wegner, 1998), on the other hand, suggests that machines that interact with even a completely symbolic world (e.g. where the operation of the program is affected by a user's key presses) cannot be considered to be Turing-equivalent. He devises an extension to the machine, which he calls an interaction machine, and shows that it also exhibits a kind of Gödel incompleteness. However, this incompleteness is shown to allow an expressiveness (representational power) that would not otherwise be available. Another extension to the Turing machine is the universal quantum computer (Deutsch, 1985). Though it can still only perform recursive (self-referential, computable) functions, it can perform them much faster (i.e. execution time scales better) than would be possible classically thanks to quantum parallelism. Kieu goes further, suggesting that—using a different formulation of quantum computation involving adiabatic processes—the Halting problem might be soluble (Kieu, 2002). However, he has so far made only limited progress in demonstrating that this is true, and even this is dependent on physical issues, the answers to which are as yet unknown.

Despite these approaches, the dominant view among philosophers, those who study cognition, and researchers in artificial intelligence is that cognitive processors can, today, only be modelled on digital computers. This is particularly true of Penrose (Penrose, 1994) who dismisses analogue computation in a couple of pages. Despite acknowledging work such as that on computation over the reals (see e.g. (Blum et al. 1989)), he claims that
there is no physical basis for non-computable functions (an argument he does not develop). Likewise, in his book on mind as machine (Crane, 1995), Crane does not even consider that analogue computation is worth a paragraph.

This view of the insignificance of analogue is represented not just in Al-related research, but in computer science generally. For instance, Dewdney (Dewdney, 1993), unusually, does have a chapter on analogue computation, but does not consider any of the issues raised in the last chapter. Instead he talks about toy problems solved by physical computers (such as sorting spaghetti length by pushing a bundle against a flat surface). Further, he suggests that analogue computation may not be able to solve any non-Turing problems, though at the end of the section he leaves the subject open. Another example is Walker's book on the Limits of computing, which has no discussion of a computer's interaction with its environment (Walker, 1994): this despite having a short section on artificial intelligence. Walker sees conventional computing as being solely a symbolic-data-driven activity: a view that makes sense in understanding the technology's limits in terms of dealing with the outside world. However, he does not point this out as a limit, nor does he consider ways of pushing back these limits (like the addition of sensors and actuators).

This mistrust or misunderstanding of analogue computation is prevalent in the literature, and can be seen in a comprehensive review of the arguments both for and against presented in the journal Behavioral and Brain Sciences (van Gelder, 1998). In Van Gelder's target article he proposes the dynamical
hypothesis: that cognitive processes are best described by dynamical systems. In his paper he goes through the arguments of what he perceives as his opposition (those who espouse the computational hypothesis, that such processes are best described as computation). Among these, one in particular is notable for our purposes (most relate to the study of cognitive science rather than its intrinsic nature or implementation). Van Gelder calls this the 'false opposition objection'. This asserts that computers and dynamical systems are in some deep way the same: either because computers are dynamical systems, or because dynamical systems are computers (and nothing more), or because dynamical systems are computable. If they are the same, then the dynamical hypothesis is either unnecessary or trivial. Using the Turing-machine definition of a computer, this would mean that dynamical systems are Turing-equivalent. Van Gelder's contention that this view is widely held is supported by the fact that many of the responses to the target article did in some way reflect this objection.

In fact, comparing computers with dynamical systems, where computers are Turing machines, is inappropriate. When Turing wrote his original paper describing the thought experiment that became known as the Turing machine, he explicitly constrained the input and output to be symbolic. Thus, the machine is unable to interact directly with its environment: it must always receive a set of symbolic inputs from sensors, which have to perform some type of transformation into the relevant symbol space (such as digitisation), and must always produce its output as a set of control symbols that can then
be used by actuators to affect the real world. This fact is critical because it means that a Turing machine cannot be considered a dynamical system (or a behavioural intelligence) that responds to some external force or signal by moving or changing state. Alone, it can only perform the intelligent manipulation of symbols. It is, therefore, not just wrong but essentially meaningless to speculate on the ability of Turing machines to perform tasks requiring human-like intelligence.

In any machine designed to interact with the environment, the outer shell (body, sensors, actuators) must be analogue. Signals from the outside world are analogous to the real physical values they represent (or, more precisely, they are the real values). For the machine to work, at some stage after this outer-shell has been breached, an analogue-to-digital (A/D) or real-to-symbolic conversion step must take place, thus allowing the digital computer to do whatever processing is required. The same, in reverse, is true for actuation. Given this, machines that can interact with their environments are hybrid: part analogue, part digital. Thus, they cannot be Turing machines.

Johnson-Laird (Johnson-Laird, 1993) hints at this in his book on computing and the mind without once mentioning the word analogue. Instead, he talks about the 'physical situation' of the limbs and about feedback loops. In a sense, his view (like that of many others) could be interpreted as saying that the body is analogue but the mind/brain is a computer and (by common definition) therefore digital. However, McLennan's argument is more
typical (MacLennan, 2001). When considering what the brain does, he considers it as a processor in isolation: he does not consider the mechanisms required to feed information in or out of the brain as part of the system.

3.2 Computational power of hybrid machines

If machines that interact with the environment are not Turing machines but hybrid or analogue machines, what is their computational power? In general, the proponents of analogue, optical, highly-interconnected, mechanical, or dynamic approaches to the engineering of intelligence do not make explicit claims that their systems are computationally superior to conventional approaches. In some cases see (Chua, 1998) for the cellular neural network or (Naughton, 2000) for optical processing they have shown that their technology is Turing-equivalent.

However, theoretical work has been done on the comparison between computers that work with real-valued inputs and the symbol-driven Turing machine. For instance, Hava Siegelmann and others have worked with notions of super-Turing or hypercomputation: forms of computation that can perform functions theoretically impossible with conventional Turing machines, such as functions based on non-computable algebras or that are non-recursive (see e.g. (Copeland, 1997)). Siegelmann demonstrated that, if allowed to take continuous rather than discrete weights, recurrent neural networks (where the outputs of neurons loop back to form the inputs of neurons that may be ‘earlier’ in their iterative chain) could perform functions that are theoretically
impossible using Turing machines (Siegelmann, 1999). Exactly what functions can be performed is dependent on many things: these include the architecture of the network, the properties of the individual neurons, and how much noise there is in the system.

Though Siegelmann is one of relatively few people studying the super-Turing or non-Turing nature of these networks, several others have investigated how noise degrades their performance. Several papers (eg. (Maass, 1997) suggest that particular Turing-computable functions (such as recognising certain types of computer language) are no longer computable in certain analogue neural networks given a certain type of noise. This suggests to some that Siegelmann's super-Turing model is physically impossible and therefore irrelevant.

The problem with this model is that it involves mathematical speculation without an interface with physics. Assumptions Siegelmann makes (about continuity, for instance, or noise) have no clear way of being validated for real hardware. Likewise, studies that look at how noise can degrade the abilities of recurrent analogue neural networks (RANNs) (Maass, 1997) do so from a theoretical rather than physical perspective. Here, the physical implementation and noise sources are not specified, so they are only hypotheses on what noise in certain kinds of systems could do (rather than what noise in real systems actually does).

Hadley (Hadley, 2000) also questions the ability of RANNs to perform Turing-equivalent (never mind super-Turing) computation. He does so on the
basis that the representation of real weights has severe difficulties because of the reliance on infinite precision: a problem that is, again, thought to come from physical realization of such machines rather than their theoretical construction. Cotogno recently wrote a more thorough paper (Cotogno, 2003) on the same issue, this time considering the possibilities of physical computation as the main theme. His concerns also rested on the issue of infinite precision but, unfortunately, he did not consider the impact of actual physics (he just made assumptions about the nature of physics). For this reason, his conclusions are only correct if his assumptions were correct. As we shall see in Chapter 7, it is not yet possible to determine this one way or another. Gandy, on the other hand, shows how important such assumptions are. In his treatment of the subject (Gandy, 1980), he shows that physical machines can only be considered to be restricted to Turing abilities in certain circumstances: if these constraints are weakened, then the machine may compute more.

The link between the theoretical and the physical is important because the brain is a recurrent analogue neural network. If Siegelmann's conclusions were applied to the brain, then Roger Penrose's (Penrose, 1989) assertion that human-like intelligence could not be performed on Turing machines could be correct (though for the wrong reasons).

Siegelmann's mathematical results can be interpreted in different ways. Consider, for instance, the Venn diagrams in Figure 3.2. Part A shows our understanding of a continuous or super-Turing machine's representational
abilities. The outermost set or superset shows what the super-Turing machine can compute (everything the Turing machine can do, and more). Inside that is the circle that shows what the Turing machine itself can do. Within that are specific Turing-computable functions. In the presence of noise, we are told that the power of the analogue network is diminished, because we can point to specific classes of Turing-computable functions that it can no longer solve. Some (such as Maas) have come to the conclusion that the computational ability of the network has now shrunk to be a sub-set of that of the Turing machine (see B). However, there is another possibility (see C): that the Turing abilities of the network are eroded, but at least some of the non-Turing abilities remain intact. In this scenario, the analogue neural network has abilities that overlap with the Turing machine: neither is encapsulated by, nor encapsulates the other. What this would mean, effectively, is that neither type of machine is particularly better or worse than the other, they're just different.

Since all real machines are—as discussed earlier—hybrid machines, then the Turing 'deficiency' becomes moot. To prove that AI was impossible in computers with sensors and actuators, even using an entirely conventional (sensor, processor, actuator) architecture and digital computation, Penrose would have to prove that analogue elements in the system add no additional computational functionality to the machine.

If analogue's computational benefits (beyond the ability to interact with the outside world) survive, do they really matter? This question is difficult to
Figure 3.2: Hava Siegelmann (A) has shown that, under certain conditions, recurrent analogue neural networks are super-Turing (can do everything a Turing machine can do and more). What happens in the presence of noise is more controversial. Two options are possible because we know that some Turing abilities are lost. (B) The network could become less powerful than the Turing machine. (C) The network could retain some non-Turing functions.
answer for two reasons. First, though the Turing machine is very well understood—all of its computational abilities have been mapped and re-mapped to produce an entirely-known landscape the concept of a non- or super-Turing machine is not. Siegelmann says that, because the study of non-Turing-computable problems is neglected, there are few available for testing (personal communication). This may be because the number of such problems is inherently low, or it may simply be that we have only found a small fraction of them so far. If mathematicians cannot or do not test for non-Turing properties then, logically, any computational structure will be found to be Turing-equivalent or sub-Turing: there are no other options. The CNN discussed in the last chapter is an excellent example of this. A study was performed to determine whether or not the system was Turing-universal, and it was found that it was (Chua et al. 1993). However, according to one of the paper authors (Tamás Roska, personal communication) they did not check to see whether it might be super-Turing. At the time, it had not occurred to them to do so, nor would they have known how to do it.

3.3 Physical computation and computing physics

If we follow the path that says that analogue computation buys us nothing in terms of additional functionality (aside from the interface with the physical world), then we are left with Penrose’s original contention that there are things that computers can’t do that brains can. Penrose (particularly in (Penrose, 1997) attributes the brain’s additional functionality not to continuous processes
but to quantum ones. His claim is that the microtubules in the brain perform quantum computations that are responsible for, among other things, consciousness. There are two main arguments against this. First, Max Tegmark (Tegmark, 2000a, 2000b), claims that the timescales related to quantum decoherence in microtubules are too fast to relate to brain processes. He suggests that classical (rather than quantum) physical description must therefore be responsible for neural behaviour. Also, work in quantum computation (eg. (Lloyd & Braunstein, 1999)) is progressing: researchers believe that they will eventually be able to construct a universal quantum computer over continuous variables. If quantum computation is genuinely important to the problem of intelligence, then we may soon have an additional tool to tackle it.

If Tegmark is right (quantum processes are not important), LaForte is right (deficiencies in Turing machines are not functionally important, and may in fact reflect deficiencies in human intelligence), and Siegelmann is wrong (RANNs are not super-Turing) then a basic question is left. Could a digital computer simulate a brain in principle? In other words, regardless of the efficiency (or not) of digital computers, are such machines capable of performing the necessary functions to replicate brain processes? Given that the brain is a physical object, it is constrained by the laws of physics. Since brains are complex and (as yet) poorly-understood physical objects, then what we are really asking is to what extent Turing machines can simulate physics. By this we mean: can they accurately predict trajectories of physical systems
over time given the precise initial conditions? Or, instead, is it true to say that physics is Turing-universal: that there are no physical functions that cannot be performed by Turing machines in finite time?

Poincaré (Poincaré, 1898) pointed out that even something as apparently well behaved as our solar-system is, in fact, not stable over time thanks to issues such as energy dissipation. Assumptions built into models that provide very good short- and medium-term predictions, can, in fact, provide answers that are qualitatively wrong in the longer term. Considering the planets as point masses, assuming masses are solid, etc., are examples of such assumptions. The physical computation that takes place in the solar system allows a myriad of factors to be taken into account simultaneously and in a distributed way. The planets do not calculate where they should be, they simply end up there based on the forces on them, their energy, mass, and so forth. Thus they end up in the right place despite system complexity. Not only is this 'computation' distributed (a truly parallel process), but it is also the ultimate analogue process in that the 'information' being processed (in terms of changing forces etc.) is not just like what it represents, it is what it represents. In other words, it is not symbolic and therefore requires no forced mapping from the real world to the mathematical world and back again in order to work.

When considering this issue, it is important to know which is more important, the physical computation, or the model. For instance, if you are using the position of the stars to find your way home based on a star chart, the
model is more important than the stars are. Thus, if there are factors that were not taken into consideration when the chart was made (the thickness of the atmosphere or shape of the planet, for instance), the movement of the stars end up performing the wrong computation. This is true of analogue systems. If the model (the numbers being crunched and the exact function performed) is more important than the hardware, then analogue is the wrong approach. The result is that, in general, digital computation makes sense if one needs to work with numbers. Scheutz (Scheutz, 1999) demonstrates this by using what is essentially a conventional Shannon analysis (relying on noise) to show that physical systems are as restricted as symbolic systems in performing computations. Vergis and his colleagues take a line related to the complexity of certain types of numerical problems (such as graph connectivity) and how such problems scale in analogue systems (Vergis et al. 1986). As these are problems where the answer lies in the mathematical domain and not the physical, it is perhaps not surprising that analogue computation does not fare well.

So, bringing us back to the question of whether or not the brain is Turing-equivalent, let us consider the perfect physical simulation as a thought experiment. First, let us assume that we know all the laws of physics and the precise current state of the universe. Next, let's grant that all this information can be programmed into an ideal Turing machine. We run the simulation. If the machine can predict the future of the universe, perfectly, forever, then physics is Turing universal. To make things simpler, in our thought experiment,
the universe consists only of classical physics: possible quantum non-determinism is not an issue. So, could the simulation work?

As we will demonstrate in Chapter 6, the answer is 'yes' only if the representational ability of the Turing machine—the fact that it can only work with Rational rather than Real numbers—does not matter. As is well known (see eg. (Stewart, 1989)), absolute precision, even to this level, matters when the functions being performed are chaotic. An assertion that the universe is Turing computable is an assertion that there are no chaotic functions implemented in the universe.

Alan Turing himself expressed this view (Turing, 1950). First he makes it clear that his model defines the limits of what any symbolic, digital computer can do.

"The special property of digital computers, that they can mimic any discrete-state machine, is described by saying that they are universal machines. The existence of machines with this property has the important consequence that, considerations of speed apart, it is unnecessary to design various new machines to do various computing processes. They can all be done with one digital computer, suitably programmed for each case. It will be seen that as a consequence of this all digital computers are in a sense equivalent."

Then he points out how very different discrete-state machines are from ordinary physical systems.

"The system of the 'universe as a whole' is such that quite small errors in the initial conditions can have an overwhelming effect at a later time. The
displacement of a single electron by a billionth of a centimetre at one moment
might make the difference between a man being killed by an avalanche a year
later, or escaping. It is an essential property of the mechanical systems which
we have called 'discrete-state machines' that this phenomenon does not occur.
Even when we consider the actual physical machines instead of the idealised
machines, reasonably accurate knowledge of the state at one moment yields
reasonably accurate knowledge any number of steps later."

3.4 Physical implementations of Turing machines

In the discussion of Van Gelder's work earlier, a thread was left unpulled: the
idea that dynamical systems (or physical systems) and computers may be the
same thing. Real computers (as opposed to those in Turing's thought
experiment) are physical objects just as brains are, and subject to the same
physical laws. How, then, can it be that computers are limited in the way that
brains are not?

The difference between physics and (classical) computation lies in
interpretation. For example, when the letter A is typed into an laptop
computer, how hard the key is pressed (within a range) is irrelevant. Soft A
and hard A are considered to be identical inputs. Likewise, the brightness of
the screen is irrelevant to the meaning of the letter A when it appears before
the user: whether or not our laptop is in power-saving mode does not affect
our perception of the output. So, lots of different physical scenarios are
counted as being identical in the computational sense. Van Gelder explains it
clearly: dynamical systems implement classical computers. That doesn't limit them to being classical computers. Whereas the physics of a device is entirely dependent on implementation and environment, the ability of a Turing machine to compute is independent of its physical substrate. Whether electronic (see, e.g. (Keyes, 1987)), optical (see, e.g. (Smith et al. 1985)), mechanical (see, e.g. (McNie et al. 1999)), or otherwise, implementation is irrelevant to the operation of the machine as a computer. It is the symbolic interpretation of the physics that matters.

Likewise, even if our Turing machine could somehow represent Real numbers, and our thought-experiment physical simulation could make perfect predictions, it still would only achieve part of what we need our embodied, artificially-intelligent agent to do. This point was succinctly outlined by Dreyfus (Dreyfus, 1972): "The formalization of a physical process is not the same as the process itself."

3.5 Other objections to AI

Before closing this chapter looking at computational power, let us briefly consider other in principle objections to the idea that we could ever build a human-like artificial intelligence. Susan Greenfield (Greenfield, 1997) has also objected to the idea that computers can do what brains do. Two of her three main objections relate to a misunderstanding of the state of the art in computing technology and specific areas of research. First, she says the number of connections in the brain is vast compared to the number of wires
available to connect processors on computer chips (this issue is one of engineering, and was considered in Chapter 2).

Her second claim is that the brain uses its memory in a completely different way than computers do. Related to this is her assertion that human beings are not 'programmed' in the computer sense, particularly not with facts, and so don't run according to the kind of rules or algorithms that are fundamental to Turing machines. While the brain learns and adapts to changing circumstances by changing its structure, she says, the computer has to stick with the set of rules it was 'born' with. Any process or idea that cannot be boiled down to this original set of rules, or derived from them, cannot be learned by the computer. Selmer Brinsjord makes similar arguments, and adds the accusation that robots could never understand narrative (even as they compose it) because they could never have sufficiently-interesting inner lives (see (Bringsjord & Ferucci, 1999) and references therein).

Such arguments shows a lack of understanding about the concept of artificial neural networks, and the concept of a learning rule: particularly surprising coming from Greenfield given her significant contact with artificial neural network researchers (such as Igor Aleksander (Aleksander & Morton, 1995)).

Her third criticism of AI is that biological brains are reconfigured dynamically, and that computers cannot be. This criticism will be taken up in our model of physical computation, presented in the next chapter.
Conclusions from this chapter

- Turing machines have known limitations to their computational power, and some believe these preclude the artificial implementation of human-like artificial intelligence.
- Physical systems cannot be simulated—with accurate, long-term predictions—on Turing machines if physics requires the performance of chaotic functions.
- Recurrent analogue neural networks, which have some brain-like properties, have been theoretically demonstrated to perform super-Turing functions in certain circumstances. Some of these abilities erode with noise.
- Neither the super-Turing nor noise-erosion results have been validated physically.
Chapter 4:

Analysis and model of physical computation

In the light of the discussions in the last two chapters about neuromorphic engineering and the question of the possibility of implementing embodied artificial intelligence using Turing machines, we recognized that a new model was needed. To be useful, we determined that this model should: have a clear interface with physics; have a clear application to engineering artificial intelligence in the simulation-of-behaviour sense (as opposed to the solution-of-arbitrary-mathematical-problems sense); and, allow for an understanding of the differences and relationships among various different types of machine.

In this chapter, we present such a mathematical model of a potentially-intelligent embodied adaptive agent. The model is entirely general, and applies to any physical system. However, it includes mechanisms that can be interpreted as allowing learning via experience of the environment.

4.1 A mathematical model of the physics of embodied systems

4.1.1 Basic definitions of the system and its environment

Let $\Sigma$ be an embodied system, here defined as an identifiable collection of connected physical elements that occupies a definable volume and has a collective contiguous boundary. The physical state of the embodied system at time $t$—by which we mean the mass, speed, position, charge, etc., of all its
constituent components—will be represented by the symbol $S_t$. We will define this not only as all of the properties of the system that could, in principle, be measured at a particular instant in time, but also their associated rates of change (acceleration and velocity, not just position, for instance).

Let $X$ be the space of all possible inputs, $x$, to the system. We use the symbol $x_t$ to mean the input to the system at time $t$. Physically, these enter via the system sensors: any and all parts of the system that can be changed by physical influences from the environment. In our model, these influences, which may include exchanges of energy, potential energy, mass, and so forth, form part of a multi-dimensional state-space. This is the simplest case. More generally, the sensor may map on to arrays of such points with geometric or topological properties, such as a camera image. However, this does not affect the general argument.

Note that the term sensor as we use it here is not the same as would be commonly-understood in engineering. The main difference, for us, is that it is sensitive to many different kinds of stimuli simultaneously—heat, light, and sound energy, for instance—and does not exclusively select out specific types of input. Also, in keeping with physics, energy must be conserved. So, whatever energy is absorbed by the sensor must also be evident in the sensor response, even if only as noise, heating, and so forth. As an example, reflection can be considered a very fast absorption and re-emission, depending on the circumstances. Thus, the sensor will be said to be complete:
that is, it must be able to absorb (or reflect) all of the available energy.

External physical changes caused by the embodied system, (emission of light, movement, etc.), are similarly called the actuator output, where an actuator is any part of the system that can change the environment, and is symmetrically analogous to the sensor as defined above. Let \( Y \) be the space of all possible such outputs, \( y \), of the system. The symbol \( y_t \) will mean the output at time \( t \).

The intelligence function, \( G \), is defined as the mapping that takes the input \( x_i \) and state \( S_i \) to an output \( y_{i+\Phi t} \), where \( \Phi t \) is analogous to the more commonly written \( \partial t \), or \( \Delta t \). This is because the model does not make assumptions about whether time and space are discrete or continuous, which are issues that relate to physics and must therefore be left open at this stage. Specific options are considered only in Chapter 6, and their physical implications are dealt with in Chapter 7.

We write \( G: (x_i, S_i) \rightarrow y_{i+\Phi t} \), with \( G (x_i, S_i) = y_{i+\Phi t} \)

For each unique \( S_i \) let the unique symbol \( G_i \) be defined, so that the symbols \( S_i \) and \( G_i \) are in one-to-one correspondence. Let,

\[
G_i (x_i) = G (x_i, S_i) = y_{i+\Phi t}
\]

Note that two symbols \( G_j \) and \( G_2 \) may refer to the same mapping, but the symbols are distinct. We will write \( \alpha(S_i) = G_i \) and \( \alpha^{-1}(G_i) = S_i \). We can represent \( G_i \) in diagrammatic form as shown in Figure 4.1.
Let the matter, space, and energy outside the boundaries of the embodied system be collectively called the **environment**, \( \Theta \), of the system. The **state** of the environment at time \( t \) will be denoted by \( \psi_t \). Let \( \mathcal{Y} \) be the set of all possible inputs from the system, \( \Sigma \), into the environment, \( \Theta \), and let \( \mathcal{X} \) be the set of all possible outputs of the environment, \( \Theta \), into the system, \( \Sigma \). The **environment function**, \( E \), is defined as the mapping that takes the environment input \( y_t \) and state \( \psi_t \) to the environment output \( x_t + \Phi_t \).

We write \( E : ( y_t , \psi_t ) \rightarrow x_t + \Phi_t \), with \( E ( y_t , \psi_t ) = x_t + \Phi_t \)

For each unique \( \psi_t \) let the unique symbol \( E_t \) be defined, so that, again, the symbols \( \psi_t \) and \( E_t \) are in one-to-one correspondence. Let,

\[
E_t ( y_t ) = E ( y_t , \psi_t ) = x_t + \Phi_t
\]

We will write \( \beta_t ( \psi_t ) = E_t \) and \( \beta_t^{-1} ( E_t ) = \psi_t \). We can represent \( E_t \) in diagrammatic form as shown in Figure 4.2.

**Figure 4.1:** The mapping \( G_t : \mathcal{X} \rightarrow \mathcal{Y} \), with \( G_t ( x_t ) = y_t + \Phi_t \)
4.1.2 Change and learning in the system and its environment

The model includes the possibility of the system state changing, which will be called learning. Let \( L_\Sigma \) be the system learning function.

\[
L_\Sigma: (x_t, S_t) \rightarrow S_{t+\Phi_t}, \quad \text{with} \quad L_\Sigma (x_t, S_t) = S_{t+\Phi_t}.
\]

We also define,

\[
L_\Sigma (x_t, G_t) = G_{t+\Phi_t}
\]

See Figure 4.3.
Chapter 4: Analysis and model of physical computation

Figure 4.3: The mappings $L_{\Sigma}(x_t, S_t) = S_{t+\Phi t}$ and $L_{\Sigma}(x_t, G_t) = G_{t+\Phi t}$

From our definition earlier,

$$\alpha L_{\Sigma}(x_t, S_t) = \alpha S_{t+\Phi t} = G_{t+\Phi t}$$

$$L_{\Sigma}(x_t, \alpha S_t) = L_{\Sigma}(x_t, G_t) = G_{t+\Phi t},$$

and the diagram commutes, $\alpha L_{\Sigma} = L_{\Sigma} \alpha$.

In the same way that we allowed change in the system’s state and called it learning for the system, we can also do this for the environment. Let $L_{\Theta}$ be the environment learning function,

$$L_{\Theta}(y_t, \psi_t) \rightarrow \psi_{t+\Phi t}, \text{ with } L_{\Theta}(y_t, \psi_t) = \psi_{t+\Phi t}.$$ 

We also define,

$$L_{\Theta}(y_t, E_t) = E_{t+\Phi t}$$

See Figure 4.4.
Figure 4.4: The mappings $L_{\psi_t}(y_t, \psi_t) = \psi_{t+\Phi t}$ and $L_{\psi_t}(x_t, E_t) = E_{t+\Phi t}$.

From our definition earlier,

$$\beta L_{\psi_t}(y_t, \psi_t) = \beta \psi_{t+\Phi t} = E_{t+\Phi t}$$

$$L_{\psi_t}(y_t, \beta \psi_t) = L_{\psi_t}(x_t, E_t) = E_{t+\Phi t},$$

and the diagram commutes, $\beta L_{\psi_t} = L_{\psi_t} \beta$.

The composite diagram shown in Figure 4.5 shows how the system and environment are coupled together. Here, $G_t$ can only be considered to cause an actuator output (change that may effect the environment) at time $t+\Phi t$ as a result of an immediate sensor input (physical influence from the environment). It cannot be considered to implement any kind of plan over time like commanding a robot arm to move through a particular trajectory: such a plan can only be carried out if the learning function $L_{\Sigma}$ allows it to, in conjunction with the inputs from the environment.
This is a subtle difference in the way we think about how machines work. Were the intelligence able to issue a command to be followed by an actuator over time, some controller would have to be at work in the actuator to make sure that the command was carried out. This is fine, but in our model, this controller is considered to be part of the intelligence function: therefore it can

**Figure 4.5:** The evolution of $G_i$ and $E_i$. 
only be considered within the same sensor-response framework as the rest of the system.

Instead, the 'plan' is carried out through the state, \( S_t \), changing with time. In the process of performing the intelligence function, the intelligent system may change in ways that would not be considered output: they do not affect the outside world in their own right. These will be called \textit{internal} changes, or changes of state (the altering of a voltage across a wire, charging of a capacitor, change in chemical composition, etc.). They cause the system to behave differently when the next input is received.

\subsection*{4.2 Illustrations of physical computation}

\textbf{Example 1}: Consider a sealed plastic bottle of water sitting on a table. If sound is made in the environment, the incoming kinetic energy \( (x_t) \) will cause the water will ripple and the plastic to vibrate \( (y_{t+\Phi_t}) \). The shape of the bottle will change \( (S_{t+\Phi_t}) \) with changes in external air-pressure \( (x_t) \). Incoming light \( (x_t) \) will be reflected from the bottle \( (y_{t+\Phi_t}) \) and/or be absorbed, heating the bottle up \( (S_{t+\Phi_t}) \), or perhaps releasing electrons \( (y_{t+\Phi_t}) \). Though the bottle has no sensors in the traditional engineering sense, it is still sensitive to the environment: it still has a sensor by our definition.

\textbf{Example 2.1}: Imagine a simple lego robot with rubberized wheels on a smooth flat desk in a sunlit room. Thanks to potential energy supplied by the force of gravity \( (x_t) \), the robot pushes down \( (y_{t+\Phi_t}) \) on the desk \( (E) \) and the solidity of the desk means the robot is pushed up \( (S_{t+\Phi_t}) \) at the same time. This
results in a bulging of the rubberized wheels (change of $S_t$) and a slight depression forming on the desk (change of $\psi_t$).

**Example 2.2**: As light hits the robot ($x_t$), some of it is absorbed, warming it, (change of $S_t$) and the remainder is reflected ($y_{t+\Phi_t}$), making the robot visible (change of $\psi_t$). The warmth causes the batteries to produce more current ($G_t$) and so additional torque on the wheels ($y_{t+2\Phi_t}$).

**Example 2.3**: Now, imagine the robot has a light sensor in its forward direction. It is configured so that the robot will move forward when the light received reaches a certain light level and will stop otherwise. This is done via a small capacitor that slowly leaks away current. If incoming light ($x_t$) builds up quickly enough (assuming the energy in $x_t$ is greater than that in $y_{t+\Phi_t}$ over a number of time periods), the capacitor will break down (conduct), current will flow, and the robot will move (releasing kinetic energy as part of $y_t$). If not, it won't.

### 4.3 Extension of the model to mediated interaction

In the previous sections, a mathematical model was developed to describe any embodied system. In this section the model is extended to discriminate between those engineered systems based on the sensing-processing-actuation model that has characterized much of robotics research. In particular, many such systems incorporate digital processors with various kinds of interfacing hardware that filters the input data and restricts the set of physical outputs available to the designer.
4.3.1 Machine with filtered inputs and outputs

The most obvious kind of input filter is analogue-to-digital hardware, that starts with inputs $X$, as before, but maps these onto a different set that will be denoted by $X'$. We will use the notation $F_{\text{in}}$ to represent such filters so that,

$$F_{\text{in}} : X \rightarrow X', \text{ with } F_{\text{in}}(x) = x.'$$

Outputs may also be mediated by the hardware interfacing any processing to the system actuators. As will be seen, machines may have outputs that have an effect on the environment outside of the formal actuators. Let $Y'$ be the set of outputs of the processor before they are mediated by the actuators of the system to become the outputs $Y$ that directly affect the environment. Let $F_{\text{out}}$ be the function mediating the outputs, with $F_{\text{out}} : Y' \rightarrow Y$. Let $V_{\text{t}}$ be the mapping from $X'$ to $Y'$ performed by the processor inside the system. We will call this a virtual machine (see Figure 4.6).
Figure 4.6: The filter functions, $F_{in}$ and $F_{out}$ map inputs $X'$ from the environment onto inputs $X$ of the virtual machine.

The state of the virtual machine at time $t$ will be denoted by $S'_t$. Let $X'$ be the set of all possible inputs from the input filter function, $F_{in}$, into the virtual machine, $V$, and let $Y'$ be the set of all possible outputs of the virtual machine, into the output filter function, $F_{out}$. The virtual function, $V$, is defined as the mapping that takes the environment input $x'_t$ and state $S'_t$ to the output $y'_{t+\Phi}$.

We write $V: (x'_t, S'_t) \rightarrow y'_{t+\Phi}$, with $V (x'_t, S'_t) = y'_{t+\Phi}$.

For each unique $S_i$ let the unique symbol $V_i$ be defined, so that, again, the symbols $S_i$ and $V_i$ are in one-to-one correspondence. Let,

$$V_i (x'_t) = V (x'_t, S'_t) = y'_{t+\Phi}$$

We will write $\gamma(S'_i) = V_i$ and $\gamma^{-1}(V_i) = S'_i$.

Like the system, $\Sigma$, and the environment, $\Theta$, the virtual machine has a virtual learning function, $L_V$. 

---
\( \mathbf{L}_V: (x'_t, S'_t) \rightarrow S'_{t+\Phi_t}, \text{ with } \mathbf{L}_V (x'_t, S'_t) = S'_{t+\Phi_t} \)

We also define,

\[ \mathbf{L}_V (x'_t, V_t) \overset{\text{def}}{=} V_{t+\Phi_t} \]

and \( \gamma (S'_t) = V_t \) and \( \gamma^{-1}(V_t) = S'_t \).

See Figure 4.7 overleaf.

From our definition earlier,

\[ \mathbf{L}_V (x'_t, S'_t) = \mathbf{L}_V (x'_t, V_t) = V_{t+\Phi_t} \]

and the diagram commutes, \( \gamma \mathbf{L}_V = \mathbf{L}_V \gamma \).
Figure 4.7: The mappings $L_V(x', S') = S'_{t+\Phi t}$

and $L_V(x', V_t) = V_{t+\Phi t}$.

4.4 The mathematical model of the system-environment interaction

To fully understand the nature of the interaction between $V_i$ and $E_i$, we need to consider the nature of the filter functions $F_{in}$ and $F_{out}$.

4.4.1 Real interaction

When $F_{in}$ is the identity function, with $F_{in}(x) = x'$ for all $x$ in $\mathcal{X}$, the input will be said to be unmediated. Likewise, when $F_{out}$ is the identity function, with
\( F_{\text{out}}(y') = y = y' \) for all \( y' \) in \( Y' \); the output will be said to be **unmediated**. We will define the situation where both the input and output are unmediated as **real interaction** (see Figure 4.8).

**Figure 4.8:** The real interaction case in which \( F_{\text{in}} \) and \( F_{\text{out}} \) are identity functions, so that \( V \) is the same function as \( G \), and \( L_V \) is the same as \( L_\Sigma \).

### 4.4.2 Virtual interaction

Where \( F_{\text{in}} \) and \( F_{\text{out}} \) are not identity functions, we define the interaction as **virtual**: an interaction between the environment and system that is either indirect or incomplete (see Figure 4.9). This situation describes the classic sensor-processor-actuator loop in which the input filter mapping is generally
many-to-one: the sensor maps a large set of physical values into a smaller set. This might be finite or infinite, but of a lesser cardinality. Alternatively, $\mathcal{X}$ may be of the same order of magnitude as $\mathcal{X}'$ (both may be uncountably infinite, for example) but may still not reflect all possible stimuli. For instance, a camera may be considered sensitive to light but not to sound, and therefore not to all incoming energy. Thus, if the machine's only sensor was a camera—no matter how good it was—many different inputs from the environment (same light level, different sound levels) would be mapped onto a single sensor output.
Let us now reconsider $F_{in}$ and $F_{out}$. To conserve $x$ and $y$ globally, while allowing $x \neq x'$ and $y \neq y'$, we can redefine these as:

$$
F_{in}(x_i) = x_i' + x_i''
$$

$$
F_{out}^{-1}(y_i) = y_i' + y_i''
$$

Where $x_i''$ and $y_i''$ are parts of the input and output that are not sensed/performed by the virtual machine. Let the \textit{complementary function} be defined as:

$$
V_{C}^i(x_i') = G_i(x_i) - V_i(x_i') = y_i'' + \Phi_i
$$

Since all variables within the machine must be conserved globally, we can define the \textit{complementary state}, denoted by $S^C_i$, such that:

$$
S^C_i = S_i - S_i'
$$

We can further define the \textit{complementary learning function} such that:

$$
L_{C,V}^i(x_i'', S^C_i) \rightarrow S^C_{i+\Phi_i}, \text{ with } L_{C,V}^i(x_i'', S^C_i) \overset{\text{def}}{=} S^C_{i+\Phi_i}.
$$

We also define,

$$
L_{C,V}^i(x_i'', V^C_i) \overset{\text{def}}{=} V^C_{i+\Phi_i} = G_{i+\Phi_i}(x_i'' + \Phi_i) - V_{i+\Phi_i}(x_i'' + \Phi_i)
$$

Since $G_i(x_i) = L_{\Sigma}(x_i', G_i, \Phi_i)$, and

$$
V_i(x_i') = L_{V}(x_i', V_i, \Phi_i), \text{ we get by substitution}
$$

$$
L_{C,V}^i(x_i, V^C_i) = L_{\Sigma}(x_i', G_i, \Phi_i) - L_{V}(x_i', V_i, \Phi_i)
$$
The complementary function is key to understanding how intelligence in the physical sense is different from our idea of intelligence in computers normally. With conventional computers, we consider intelligence functions on subsets of stimuli represented as abstractions (such as numbers). This makes them implementation independent. For example, an adder can be built in many different ways with many different materials. However an external observer of some kind is required to determine what this intelligence function actually is and to interpret: to determine the meaning of the results according to their own frame of reference and application. With real interaction, the function and implementation are one and the same, and are environment-dependent. A sundial does not work in a world without sun, nor an abacus in a world without gravity.

We can put this picture together with the composite diagram in Figure 4.5 to see how the model functions as a whole: see Figure 4.10.

Example 3: Consider again the lego robot discussed in Example 2. Incoming light from the environment \( (x_t) \) is mapped via sensors and analogue-to-digital conversion \( (V'_t) \) into a form \( (x'_t) \) appropriate for the processor \( (V_t) \). Changes in voltage and current, etc., within the complementary function in the physical machine \( (V'_{t+\Phi_t}) \) are instantaneously mapped to changing operation of the processor in the virtual machine \( (V_{t+\Phi_t}) \), which can then be understood purely by considering its input \( (x'_t) \) and virtual learning function \( (L_v) \). The
4.4 Real versus virtual interaction

In the real interaction case, implemented in the physical world, various constraints may be imposed by the laws of physics, \( L_p \), which we define as a
function that maps any physical state to the next physical state in time based on physical inputs. (Such a function implements the physical laws as they exist in nature, rather than as we understand them as scientists today). These include the following:

1. Quantities such as energy may have to be conserved.

2. Changing functions and variables in the model may be constrained to vary continuously or to have particular allowed values. This would prevent a system, for instance, moving from one point to another without going through those in-between.

Where the laws of physics (which we can write as $L_P$) constitute the only mechanism available for the state of the system or environment to change, we can say that:

$$L_\Theta = L_\Sigma = L_P$$

For the virtual-interaction case some major constraints are lifted: with important consequences. Since $x'_t$ is not constrained to be equal to $x_t$ (likewise for $y'_t$ and $y_t$, and $S'_t$ and $S_t$), we can say the following. First, the virtual machine need not operate on the inputs and outputs in their totality, but rather selectively: entire sensor modalities—such as sound energy or kinetic energy—may be excluded by $x'_t$ (actuator modalities by $y'_t$), or specific ranges within specific modalities may be excluded. This can be considered in two ways. From a design perspective, this means that behaviours (sensor-actuator pairs, as defined earlier) may be considered equivalent even if the global inputs and outputs (the inputs and outputs taking all possible modalities into account) are
different. As mentioned before, how hard a key is pressed (in normal operation) does not affect the operation of the computing machine within a laptop.

Second, in virtual interaction, \( E \) and \( V \) need not have any kind of conserved relationship (such as conservation of energy), and \( L_V \) need not be the same as \( L_{\Theta} \) (or \( L_{\mu} \)). Because only specific range and modality subsets of \( \mathcal{X} \) and \( \mathcal{Y} \)—subsets relating to sensor sensitivity being bounded and confined to a specific type of energy, respectively—affect \( V \) and \( L_V \), the intelligence and learning functions are partially decoupled from the environment. They can evolve in a way that is only partly driven by external inputs. If the machine and sensors are designed carefully, arbitrary choice of \( V \) and \( L_V \) may be made. To make this more concrete, under normal operating conditions, a laptop's behaviour has more to do with its software running than what is going on in the room around it: this is because it is partially shielded (decoupled) from the environment by the complementary function.

It is important to note that the arbitrary choice of virtual function, \( V \), (software) relies on the creation of virtual inputs and interpretation of virtual outputs by the complementary function, \( V^C \). This is because a real interaction with the environment (where the learning function is \( L_{\Sigma} \)) must be taking place at the same time as the virtual one. Thus, it is only because of \( V^C \) that non-physical evolution is allowed for \( V \) and \( L_V \).

Note that, to do all this, \( F_{\text{in}} \) and \( F_{\text{out}} \) are as important as \( V^C \) to allow
operation of the virtual machine. To this end, we will define the **global complementary function**, \( V^c \), as the combined \( F_{out} V^c F_{in} \).

**Example 4:** Before moving on, it is worth going back to the laptop that we mentioned earlier to compare the real- and virtual-interaction analyses of it. In the real case, the machine is just a physical object obeying the physical laws. It moves under gravity, has resistance, gives off heat, makes noise when tapped, and so forth. It can be used as a paper-weight or a doorstop. All of these uses of the machine (as we may interpret them) remain consistent with this picture. In the virtual analysis, however, only the input of the pressing of the keys and the output of the pixels on the screen are part of the machine's virtual intelligence function: \( V \). Other aspects of its behaviour, which combine to give it the global function \( G \), are performed by \( V^c \).

For simulated robots, artificial life, and other applications where the interaction is purely symbolic and has no physical output to the environment (apart for display purposes), \( V \) is all that's important: the complementary function is irrelevant (as it is normally considered in computer science). For real robots in a real world, however, \( V^c \) is critical and must be taken into account during design and fabrication. \( V^c \) performs all the functions that engineers have to consider but that computer scientists generally don't: it is the implementation of the virtual machine and its interface with the real world. However well-designed the virtual machine, sensitivity to the environment and efficiency are both dependent on \( V^c \).
Conclusions from this chapter

We have now set up a model that addresses the ability of physical objects to sense (using a wider definition of the word than is conventionally used in engineering), actuate, and learn in the most basic sense. The framework we have set up shows:

• There are two main types of interaction—real and virtual—that can take place between a machine and its environment.
• Real interaction is unmediated and allows perfect coupling between the two systems.
• Virtual interaction involves some kind of mediation: generally in the form of a symbolic interface. This removes some constraints from the intelligence function to be performed, but real interaction at the physical level is still required.

Issues raised in this chapter to be discussed further

At this stage we have not probed the model in any way: merely set it up and defined some terms. In Chapter 5 we will show how the model can accommodate noise, and in Chapter 6 we will determine the consequences of the model, assuming all mathematical possibilities (without regard to physics). Then, in Chapter 7, we will allow for these options to be reduced by taking real physics into account.
Chapter 5:
Computing machines and noise

Now that we have defined our mathematical model of computation, we must connect it to computer science and electronic engineering. We will do this by defining the machines we are used to dealing with—Turing machines, strictly-limited-precision digital computers, and analogue machines—and showing how they may be physically implemented in our model. In particular, we will consider the sets \( X \), \( Y \), \( X' \), and \( Y' \), and what they would have to be in order to support the type of computation in question.

To facilitate this, we will first review some mathematics that are important to this discussion.

5.1 Number systems and cardinality

Extending the intuition of the size of finite sets to infinite sets can lead to many paradoxes that can be avoided using the notion of cardinality. The cardinality of a finite set is defined to be the number of its elements: for example, the cardinality of the upper case alphabet is 26. The cardinality of infinite sets are defined using \( \aleph \) (the Hebrew symbol aleph) with a subscript of 0, 1, 2, etc., to describe the order or class of infinity of the set.

The natural numbers, \( \mathcal{N} = \{ 1, 2, 3, 4, \ldots \} \) have the lowest level of cardinality, \( \aleph_0 \) (pronounced 'aleph null'). This is known as a denumerable or
**countable** infinity: a term that comes from the fact that, though you would never finish counting the natural numbers, you could clearly see how to start.

Two sets are said to have the same cardinality if they can be put in one-to-one correspondence with each other. Consequently, it can be shown that the set of integers, \( \mathbb{I} = \{ \ldots, -3, -2, -1, 0, 1, 2, 3, \ldots \} \), is countable, and has cardinality \( \aleph_0 \). For example:

\[
\begin{array}{cccccccc}
-3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
7 & 5 & 3 & 1 & 2 & 4 & 6 & 8
\end{array}
\]

establishes a one-one mapping \( f: \mathbb{I} \rightarrow \mathbb{N} \) with the formula \( f(i) = (i-1) \times 2 \) for \( i \geq 0 \), and \( f(i) = 1 - 2i \), for \( i < 0 \).

This example shows that a set can be in one-to-one correspondence with a subset of itself, since \( \mathbb{I} \subset \mathbb{N} \), which may be counter-intuitive. Likewise, the rational numbers, \( \mathbb{Q} = \{ m/n \mid \text{for all } m \text{ and } n \text{ belonging to } \mathbb{I} \} \), also have cardinality \( \aleph_0 \). The proof of this is beyond the scope of this thesis.

The real numbers, \( \mathbb{R} \), are defined as all rational numbers and all irrational numbers, \( \mathbb{Q}' \), that no fraction of integers can represent. \( \mathbb{Q}' \) has cardinality \( \aleph_1 \), which means that it is an **uncountable** or **non-denumerable** set. This means that, strictly, it has more elements than \( \mathbb{Q} \) and a 1-1 mapping between the two sets is not possible. By cardinal arithmetic, the result of the
addition or multiplication of two infinite cardinal numbers is simply the larger of the two. Thus, since \( R = Q \cup Q' \), its cardinality is \( \aleph_0 + \aleph_1 = \aleph_1 \). The set of real numbers is also known as continuous (or the continuum).

The continuum is one of a number of sets that has the property that a subset defined by an interval (say between 0 and 1) has the same cardinality as the set itself. As well as \( R \), this is true for \( Q \) and \( Q' \). Thus, the cardinalities of any finite interval of these three sets are \( \aleph_1 \), \( \aleph_0 \), and \( \aleph_1 \), respectively. Proof of these concepts is, again, beyond the scope of this thesis. For further information, please see e.g. (Borowski & Borwein, 1989) or (Lipschutz, 1964).

### 5.2.1 Types of computing machine

In this section we will cover the three main types of computing machine and consider some of their mathematical properties.

**5.2.1 The Turing Machine**

A **Turing machine** is an idealised symbolic computer as defined by Turing (Turing 1936): for a full description of the way a Turing machine works, see Chapter 3. We define **Turing functions**, \( T \), as those that can be performed on a Turing machine, and comprise the set \( T \). These Turing functions operate on \( C \), the set of **computable numbers** (also as defined by Turing), which includes the rational numbers \( (Q) \) and those irrational numbers (from \( Q' \)) that can be
represented by finite means (in a finite or countable set of specialized symbols such as \(\pi, \sqrt{2}, \sqrt{3},\) etc.). From Turing's paper, and from the arguments given in the last section, we know the set of computable numbers is countable, that is, it has cardinality \(\aleph_0\). Thus, in any implementation of a Turing machine, either \(X\) or \(X'\) must be \(C\).

### 5.2.2 The strictly-limited precision digital computer

We can also usefully define a **strictly-limited-precision digital computer** as having the same operational properties as a Turing machine while being strictly limited in the number of available states available to it: that is, the number of states must be finite. The **strictly-limited-precision digital functions**, \(S\), comprise the set \(S\). Such functions act on a finite set (cardinality \(\mathcal{N}_{\text{max}}\)). Thus, in any implementation of a strictly-limited-precision digital computer, either \(X\) or \(X'\) must be \(\mathcal{N}_{\text{max}}\).

### 5.2.3 The analogue machine

We define an **analogue machine** as one that can map a set of physical values isomorphic with the continuum \(\mathcal{R}\) onto itself. The **analogue functions**, \(A\), comprise the set \(A\). For the purposes of our analysis in this thesis, we will define these functions as operating on the real numbers, \(\mathcal{R}\). Thus, in any
implementation of an analogue machine, either $X$ or $X'$ must be $R$. However, it should be remembered that for analogue machines—unlike the machines defined previously, which operate on numbers—inputs (and outputs) are exchanges of energy.

**Example 5.1**: An analogue function may be implemented via a direct physical process, such as light being focussed by a lens to produce a Fourier transform. It may also be implemented through an indirect process: incoming light may produces an intensity-dependent photo-current, which feeds into a motor, which in turn rotates with a current-dependent speed.

### 5.3 Adding noise to analogue machines

Shannon's communication theory (Shannon & Weaver, 1949) describes how two inputs from the environment may appear to be the same because of the existence of noise, and sets up a framework to quantify this. Here we will present a simplified view of Shannon's theory—ignoring considerations of bandwidth and entropy that are not relevant to our model—and show how it affects the performance of the analogue machine.

Shannon defines **communication** as the process of a sender relaying a message to a receiver via a noisy channel (see Figure 5.1).
Figure 5.1: Shannon's communication theory considers the scenario where a sender is sending a message to a receiver via a noisy channel. The noise prevents the receiver from being able to distinguish very similar messages with high certainty, thus the bandwidth (the amount of information that can be reliably transmitted) is limited.

We will explain this concept as follows. Let us hypothesize a sender (here, the environment) sends a sender's message via light energy, $x$, to the receiver (system). The input consists of a stream of photons coming in (at which time $x=1$, for instance), and the sender's message is encoded as a regular time interval, $\Delta \tau_s$, between photons (during which time $x=0$, for instance).

Now, let us hypothesize that $F_{in}$ is a noisy channel: that is, it maps the constant interval, $\Delta \tau_s$, to an observed interval, $\Delta \tau_r$, that has a normal distribution. In order to make our treatment as general as possible, we will consider two possibilities for the $F_{in}$ mapping and subsequent measurement: that the measurements are dependent on each other or they are not. Though the second option is more mathematically complicated, it more closely matches the physics we are trying to model (discussed in Chapter 7) and so is important to describe in detail here.
5.3.1 Dependent measurements

Consider the situation where the noise means that time is added or subtracted from the original interval $\Delta \tau_s$, but always relative to the original signal: that is, if a photon arrives 'early' at the end of one $\Delta \tau_r$, the time of the next interval will be longer than it would have been otherwise. This situation is illustrated in Figure 5.2 below.

[Diagram showing time intervals $\Delta \tau_s$ and $\Delta \tau_r$, with marks indicating photon arrivals at A, B, and C.]

**Figure 5.2:** Here, a message encoded as a time interval between photons, $\Delta \tau_s$ (grey marks below axis), with noise mapping the sender's signals to new, noisy intervals $\Delta \tau_r$ (black marks above axis). The letters A, B, and C, mark three specific time intervals, $\Delta \tau$, over which the noisy signal may be received.

Now consider counting photons over the time intervals $0 \rightarrow A$, $0 \rightarrow B$, and $0 \rightarrow C$. In the first of these, a photon that should not have been counted arrives early: thus what should have been a count of one is two instead. In the second, a photon that should have been counted arrives late: thus the receiver counts eight photons instead of nine. Finally, in the interval $0 \rightarrow C$, photon 13 arrives early, and thus is counted when it should not be.

Notice that the noise in the middle makes no difference to the accuracy of the overall photon count: only the noise at the ends matters. At worst, this means the photon count will be out by two for any particular measurement.
(the first arrived too early to be counted, the last too late, or the first arrived late and so was counted, and the last arrived early). Thus, the number of photons received is $\Delta t \pm 2$. Let us assume that the mean of the noise is 0. As a fraction of the total received signal, the maximum error can be written as: $2\Delta t / \Delta t$. Since the top line is fixed, this maximum error reduces with the time interval over which the measurement takes place: thus the precision increases with time.

5.3.2 Independent measurements

Next, let us consider that $F_{in}$ performs a different kind of noisy mapping. Here we can say the 'clock is reset' between measurements and the length of one interval is completely independent of the length of another. In such a case, a possible (though improbable) situation could involve many short intervals occurring in a row, resulting in an accumulated error.

Here, we must consider the normal distribution in more detail. It has a standard deviation $\sigma$, variance $\sigma^2$, and population mean, $\Delta t_{i, pop}$, for all messages, such that:

$$\Delta t_{i, pop} = \sum_{i=1}^{N_p} \frac{\Delta t_i}{N_p} = \Delta t_s$$

where $\Delta t_{i,}$ is the value of the $i$th measurement (the time between the $i$th and $(i+1)$th photon). $N_p$ is the total number of possible measurements in the
distribution: which is the duration of the light signal divided by $\Delta \tau$. If the light signal goes on indefinitely then $N_p$ is arbitrarily large but finite.

For a sample that has a distribution (rather than a single value) that is mono-modal and symmetrical (which includes normal, Gaussian, and local white-noise distributions) the population mean is the same as the value being measured without noise (see Figure 5.3). Over any period of time, $\Delta t$, the interval $\Delta \tau$ is accumulated in the state of the system $S_t$, resulting in a sample mean, $\bar{\Delta \tau}^w$, where:

$$
\bar{\Delta \tau}^w = \frac{1}{N_S} \sum_{i=1}^{N_S} \frac{\Delta \tau_i}{N_S},
$$

and $N_S$ is the number of measurements in the sample. The sample mean is defined to be the receiver's message at time $t + \Delta t$, where $t$ is the time of the initiation of the signal. This is illustrated in Figure 5.2.

**Figure 5.3:** Here, a message encoded as a time interval between photons, $\Delta \tau$, is mapped over time $\Delta t$ to a normal distribution of received intervals $\Delta \tau_r$. 
5.3.3 Confidence for independent measurements

Where noise causes the arriving message to be distorted, the number of possible unique messages that can be sent reliably is reduced. This is because each one must be distinguishable from the others with a given confidence: defined as the probability that the message arriving is the same as that intended by the sender.

Because the former is dependent on the latter, we will consider the meaning of confidence in detail here, distinguishability in the next section. This requires first defining another statistical term: the standard error of the mean, commonly known simply as the standard error, is defined as the error for a given sample \((\Delta r^N_t - \Delta r^{\text{pop}}_t)\), and is an indication of how close the population and sample means are to each other. Written as \(\sigma_m\), it is defined as follows:

\[
\sigma_m = \frac{\sigma}{\sqrt{N_S}}
\]

Note that the standard error gets smaller as the number of measurements increases (and so as \(\Delta t\) increases).

Finally, we can define the confidence level, \(\chi\), of a particular \(\Delta r^N_t\) that varies inversely with the standard error. Expressed as a percentage, this is the probability that the actual value of a measurement lies within given bounds. The set of values that fall within these bounds is defined as the confidence interval. We will define measurement precision (as opposed to statistical
precision, which is related to the variance) as varying directly with the **width**, \( w \), of the confidence interval for a given confidence level. This is illustrated in Figure 5.4.

![Figure 5.4](image)

**Figure 5.4**: Here, we show how the width of the confidence interval, \( w \), is defined for a given confidence level, \( \chi \).

Thus, for a given \( \chi \), where \( 0 < \chi \leq 1 \), there exists a neighborhood \( \Delta \tau_s \pm w/2 \) such that:

\[
\text{probability (|\( \Delta \tau_r - \Delta \tau_s \)| < \( w \)) > \( \chi \)}
\]

It follows here that, as \( w \) increases, \( \chi \) increases. Also, as \( \Delta t \) increases, \( w \) decreases (and precision increases) for a specific \( \chi \), or \( \chi \) increases for a specific \( w \).

Thus, as in the case of the dependent measurements, precision increases over time. However, here it increases (error decreases) more slowly, in that the error is proportional to \( 1/\sqrt{\Delta t} \), and not \( 1/\Delta t \).
5.3.4 Distinguishability for independent measurements

Now we can properly define *distinguishability*. Consider $\Delta \tau_{s_1} \neq \Delta \tau_{s_2}$ with $|\Delta \tau_{s_1} - \Delta \tau_{s_2}| < w$ for some chosen confidence level $\chi$. We define $\Delta \tau_{s_1}$ and $\Delta \tau_{s_2}$ as therefore being *indistinguishable* at that confidence level (see Figure 5.5).

![Figure 5.5](image)

**Figure 5.5:** Where $\Delta \tau_{s_1} \neq \Delta \tau_{s_2}$ and $|\Delta \tau_{s_1} - \Delta \tau_{s_2}| < w$, we cannot distinguish between two values.

We can now choose an arbitrary $x' \in X'$, and construct a set of points such that:

$$D = \{ x' \pm nw \mid n = 0, 1, 2, \ldots \},$$

We define this as the set of *distinguishable values* for a particular $w$ and $\chi$. Note that, because $D$ depends on $w$, it also depends on $\Delta t$. The maximum error at this confidence level—half a confidence interval—can be reduced only by increasing the number in the sample (time taken). By construction, the cardinality of $D$ is countably infinite.
We can generalize from the analysis above to all distributions where the sample mean tends to the population mean with increasing $N$. This would include all monomodal distributions that are symmetrical about the mean, including bounded *white noise* distributions (defined as random distributions to a local neighborhood).

**Example 5.2** Say we wanted to quantify intensity using a measurement with a $\chi$ of 99% and a $w$ (precision) of 0.1 candela. Here, we might consider that the brightness in candelas could be adequately represented using just one decimal place, even though a given measurement itself might have had many decimal places. Thus, we can think of the intensity as being cut up into grey levels of 0.1 candela each. For a given intensity range, say 0-5 candela, this would mean there were 50 distinguishably different intensity values. If the confidence interval were reduced (by increasing the time of the measurement), then both the number of decimal places and possible grey levels would increase for the same confidence level.

**5.4 Perfect versus noisy analogue machines**

From the arguments above, we can now split the analogue machines into two groups. Where noise is not present, and $F_{in}$ performs a 1-1 mapping from $\mathcal{R}$ to $\mathcal{R}$, we will call them *perfect analogue machines*. Where noise is present, and $F_{in}$ performs a many-1 mapping with a distribution compatible with the analysis above, we will call it a *noisy analogue machine*. In this case we will
say that it operates not on the reals, but on the distinguishable values or intervals $\mathcal{D}$.

**Conclusions from this chapter**

• Different types of machine operate on different sets with different cardinalities.

• Turing machines operate on the set of computable numbers, $\mathcal{C}$, which is countable.

• Strictly-limited-precision digital computers operate on finite sets.

• The sets that analogue machines can be said to work on varies depending on the presence or not of noise. Without noise, the set is uncountable. If the noise has a normal distribution, the set is $\mathcal{D}$, the distinguishable values, which is countable.
**Chapter 6:**

**Consequences of the model**

We have set up a model that shows how a machine can simultaneously be considered both a physical object and a machine performing a computation. This has been done without making any assumption about whether the environment is bounded (or not), or that space-time is continuous (or discrete), or considering noise. In this chapter we will consider how the specific details of computing machines and the sets they operate on are related to the physics under which they are implemented. By doing this, we can clarify the specific representational abilities of the intelligence function in different physical scenarios, and how these relate to behaviour. In Chapter 7, we will discuss the validity of these physical scenarios.

**6.1: Do machines differ in their ability to adapt to the environment?**

Here we consider the difference between different types of machine in different mathematical scenarios. To answer this question requires that we look at the following sub-questions:

1. What constraints might physics place on the environment and intelligence functions \((E \text{ and } V)\) and the domains they operate on? (Note, we will always assume a machine is virtual unless we can show otherwise.)
2. How does the physical interaction class (real or virtual) differ with the various machine types?

3. What is the information retention for the different machines in the different physical scenarios?

4. Can virtual interactions approximate real interaction?

**Question 6.1.1 What constraints might physics place on the intelligence and environment functions and the domains they operate on?**

Here it is necessary to consider the various possibilities for \( \mathcal{X} \) and \( \mathcal{X}' \), each of which has computational and physical interpretations. Mathematically, each of these options will be considered in this analysis without any sense of whether or not they are correct in physical terms (though we will suggest what they might mean in physical terms). The physical validity (or not) of these options is considered in Chapter 7.

**Assumption:** For the remainder of this analysis, we will assume that there are only four types of machine that \( E \) and \( V \) can be: the perfect analogue machine, the Turing machine, noisy analogue machine, and the strictly-limited digital computer (as defined in Chapter 5).

From this it follows that there are four options for \( \mathcal{X} \) and \( \mathcal{X}' \):

1. \( \mathcal{X} = \mathcal{R}, \ |\mathcal{X}| = \aleph_1 \) and/or \( \mathcal{X}' = \mathcal{R}, \ |\mathcal{X}'| = \aleph_1 \)

2. \( \mathcal{X} = \mathcal{C}, \ |\mathcal{X}| = \aleph_0 \) and/or \( \mathcal{X}' = \mathcal{C}, \ |\mathcal{X}'| = \aleph_0 \)
3. $X = D$, $|X| = \aleph_0$ and/or $X' = D$, $|X'| = \aleph_0$

4. $X = N_{\text{max}}$, a finite set, $|X| = n$ and/or $X' = N_{\text{max}}$, $|X'| = n$

The type of element in each set is important: analogue machines operate on physical values isomorphic to the reals or distinguishables, digital computers and Turing machines operate on numbers or symbols. However, for this discussion we will focus on cardinality rather than the actual sets themselves because we are interested in how they map from one to another. This is because we are particularly interested in whether a 1-1 mapping exists and a real interaction, as described in Chapter 4, is possible.

In the next section we will consider what each of the four possibilities that result from our assumption above mean physically. There is some circularity the discussion in that the abstract definitions of machines come (in part) from physical definitions, and we are here simply mapping back to the latter. Nevertheless this step is important because it ensures the link between the mathematical, computational, and physical is unambiguous.

Option 6.1.1.1:

$X = \mathbb{R}$, $|X| = \aleph_1$ and/or $X' = \mathbb{R}$, $|X'| = \aleph_1$

By definition, this means $E$ and/or $V$ is a perfect analogue machine. Mathematically, this means that there is an uncountably infinite number of possible states that the environment (or system) can map to and from. Physically, this would have to mean that at least some aspect of the universe—
some physical parameter that might change over time—is continuous. As shorthand, we will call this parameter **space-time** in this chapter. However, in Chapter 7 we will consider in more detail the nature of continuity/discontinuity in physics.

**Option 6.1.1.2**

\[ X = \mathfrak{C}, \ |X| = \aleph_0 \quad \text{and/or} \quad X' = \mathfrak{C}, \ |X'| = \aleph_0 \]

By definition, this means \( E \) and/or \( V \) is a Turing machine. Here, the environment/intelligent system is has a countably-infinite number of potential inputs/outputs which means space-time is not continuous (we will call this is **discretized** space-time).

**Option 6.1.1.3**

\[ X = \mathfrak{D}, \ |X| = \aleph_0 \quad \text{and/or} \quad X' = \mathfrak{D}, \ |X'| = \aleph_0 \]

By definition, this means \( E \) and/or \( V \) must be a noisy analogue machine. Physically, this would mean that space-time is continuous but an intrinsic physical noise exists. A full argument is given in Chapter 7, but here we will say that intrinsic noise is an indication of quantum non-determinism. Note that since \( F_{in} \) must, by definition, exist for noisy analogue computation (as described in Chapter 5), real interaction cannot take place between a noisy analogue machine and any other machine: the interaction must be virtual.

**Option 6.1.1.4:**

\[ X = \mathcal{N}_{\text{max}}, \text{ a finite set,} \quad |X| = n \quad \text{and/or} \quad X' = \mathcal{N}_{\text{max}}, \quad |X'| = n \]
By definition, this means $E$ and/or $V$ is a strictly-limited-precision digital computer. The physical interpretation of this is that the inputs/outputs of the environment and/or system are finitely discrete and bounded.

**Question 6.1.2: How does the physical interaction class (real or virtual) differ with the intelligent machine types?**

This question, too, can be broken into parts:

1. Can real interaction exist (or is the environment a noisy analogue machine and the universe non-deterministic)?
2. Is the simplest condition for real interaction—$\mathcal{X}' = \mathcal{Y}'$—met?
3. If not, is the $F_{\text{in}}$ that performs the 1-1 mapping between sets the identity function? If no, then the machine must be virtual.

We will use the following assumptions throughout the remainder of the analysis. The physical validity of these assumptions will be considered in Chapter 7.

- **Assumption 1:** The domain and range of $V$ are the same: $\mathcal{X}' = \mathcal{Y}'$. This is also true for $E$: $\mathcal{Y}' = \mathcal{X}$.

- **Assumption 2:** The domain/range of the environment reaction function, $E$, is either a superset of the domain/range virtual machine, $V$, or the same set. That is:

$$\mathcal{X} \subseteq \mathcal{X}', \mathcal{Y} \subseteq \mathcal{Y}'$$

The condition for real interaction, as we saw in the last chapter is:
∀x, x=x′

For this to be possible, then: X=X'=Y=Y'

Question 6.1.2.1: Can real interaction exist (or is the environment a noisy analogue machine and the universe non-deterministic)?

This question depends only on the nature of physics (the controversial area of quantum mechanical interpretation). For this reason, it will not be considered until our discussion of physics in Chapter 7.

Question 6.1.2.2: Is the simplest condition for real interaction met?

Obviously, in the following three cases, interaction is real:

\[ X = \mathcal{R} \quad \text{and} \quad X' = \mathcal{R}, \]

\[ X = \mathcal{C} \quad \text{and} \quad X' = \mathcal{C}, \]

\[ X = \mathcal{N}_{\text{MAX}} \quad \text{and} \quad X' = \mathcal{N}_{\text{MAX}}. \]

Question 6.1.2.3: If not, must \( F_{\text{in}} \) exist to perform a 1-1 mapping between sets?

There are two cases where the simplest condition for real interaction is not met, yet there can be a 1-1 mapping between \( V \) and \( E \). These are the cases where:

\[ X = \mathcal{C} \quad \text{and} \quad X' = \mathcal{D}, \quad \text{or} \]

\[ X = \mathcal{D} \quad \text{and} \quad X' = \mathcal{C}. \]
In these cases, the condition for real interaction, is violated:

\[ \forall x, x = x' \]

unless we can show that \( C = D \).

Though they are sets with the same cardinality, \( C \) and \( D \) have members of different types (the former consists of numbers, the latter of physically-expressed quantities). This mismatch in the type of element in the two sets precludes the possibility of real interaction in our model: some \( F_{in} \) is required to perform the 1-1 mapping from one into the other.

Further, this will always be true of a Turing machine or a derivative of it: such as the strictly-precision-limited digital computer. Since they all require physical values to be mapped into numbers, all require a filter function \( F_{in} \) that is not the identity function. This is a re-statement of the fact that Turing machines have no mechanism through which to directly interact with the physical world.

Finally, even if the above were not the case, noisy analogue functions—as we have defined them—are inherently virtual \( (X \neq X') \).

**Return to Question 6.1: How does the physical interaction class (real or virtual) differ with the various machine types?**

- If \( V \) is a Turing machine (or a sub-Turing derivative) then its interaction with the environment may only be real if the environment \( E \) is also a Turing machine operating on symbolic (non-physical) inputs and
Chapter 6: Consequences of the model

outputs.

• If $V$ is an analogue machine, its interaction with the environment may only be real if physics is deterministic.

• If $V$ is a strictly-limited-precision digital computer, its interaction with the environment may only be real if the environment is bounded and operates on symbolic (non-physical) inputs and outputs.

**Question 6.2: What is the physical information loss for the different machines in the different physical scenarios?**

In this section we will consider the interface between the environment and machine from a simplified information-theoretic perspective. We will define information in a similar way to Shannon: as a measure of the confidence (as defined in Chapter 5) we have in a particular value or message. This can be considered its probability of having been 'sent' by the environment.

Consider the mapping in Figure 6.1 below. Here, $\mathcal{X}$ has finite cardinality $m$, $\mathcal{X}'$ has finite cardinality $n$, and $F_x$ maps each $x$ onto its nearest $x'$ without noise.

As shown, for finite, discrete sets the probability that a given $x'$ came from any particular $x$ is:

$$P_x = n/m,$$

and is constant for all $x$ (in this case).
Figure 6.1: To understand the interaction between intelligence and environment functions we can look at the interface between them and the information loss that occurs. Here ten possible outputs from the environment are being mapped to two possible inputs to the intelligence. This might be a mapping of ten light intensity grey levels to two, or of two light intensity grey levels with five different audio amplitudes that are not sensed by the virtual machine.

We will define this probability as the information retention, $P$, for a given $F_{in}$, where $P = 1$ if $F_{in}$ performs a 1-1 mapping.

Again, though this is similar to Shannon information in many ways, it is not the same, for the same reasons that were discussed in Chapter 5.
This ties in with our model in the sense that real (as opposed to virtual) computation, as we have defined it, involves full retention of physical information. Further, it gives us a way of discriminating between different virtual machines.

For the cardinalities of infinite and continuous sets, division is not defined. However, determining the information retention is generally straightforward, in that a 1-1 mapping is either possible or not. If such a mapping is possible, we will say that the information retention is 1. If not, it is 0. From our discussion of cardinal numbers at the beginning of Chapter 5, by definition, we know that a 1-1 mapping is only possible if two sets have the same cardinality. Thus, where a set with cardinality \( \aleph_1 \) maps to a set with cardinality \( \aleph_1 \), we will define information retention as 1. For \( \aleph_1 \) to \( \aleph_0 \) or \( \aleph_0 \) to a finite set, we will define the information retention as 0. As we have previously stated, we set the condition for the model that \( X' \subseteq X \), so these are the only options possible that include infinite cardinalities.

### 6.2.1 Calculating physical information retention

Using this definition we can now calculate the information retention for different physical scenarios and machines. We will do this in three stages. First we will consider the cardinalities of \( X \) and \( X' \); both for the number of possible inputs for a single measurement—we will call these \( m_i \) and \( n_i \), where the \( i \) stands for instantaneous—and again where \( \Delta t \rightarrow \infty \). The latter will show how
the number of inputs changes as the measurements get longer, and we will call these $m_i$ and $n_i$.

The final step will be determine the appropriate value of $P$. Though splitting this up may seem artificial, we believe it will make clearer the connection between the physics and mathematics.

6.2.2 Assigning physical properties to $X$ and $Y$

To understand the set of environmental outputs, there are three physical questions we must consider: these are issues that are as yet unresolved in physics, and yet would affect the possibilities for physical information exchange. Their full physical meaning and implications will be left for Chapter 7, but the issues are as follows:

1. Space-time may be either continuous or not (discrete). We will annotate scenarios where we assume it is continuous as $C$ and use $[C]$ otherwise.

2. Physics may be deterministic, or not. We will label these two possibilities as $D$ and $[D]$, respectively.

3. The universe may be physically bounded, or not. We will call these options $B$ and $[B]$, respectively.

We will take all possible combinations of these: each one essentially representing a universe where a different type of physics exists. Since we will be moving between similar scenarios in each case, we will focus on explaining
the difference between each and those that precede it. Also, in order to avoid confusion between finite sets as defined by a designer (such as $\mathcal{X}$ the SLPDC) and finite sets that exist because of discreteness/boundedness in the environment, we will write the former as $\mathcal{N}_{\text{max}}$ and the latter as $\mathcal{N}_{\text{MAX}}$.

### 6.2.2.1 Possible scenarios for the environment

**Environment scenario CDB**

By definition, if space-time is continuous, the environment operates over the real numbers:

$$\mathcal{X} = \mathbb{R}, \ |\mathcal{X}| = \mathcal{N}_1$$  

If physics is deterministic, there is no noise and no discretization width $w$. Thus, the sets in operation remain the same:

$$\mathcal{X} = \mathbb{R}, \ |\mathcal{X}| = \mathcal{N}_1$$  

The cardinality of the reals is uncountably infinite on a line, whether bounded or not, so:

$$\mathcal{X} = \mathbb{R}, \ |\mathcal{X}| = \mathcal{N}_1$$  

Since there is nothing here to vary with time, for this environmental scenario:

$$m_i = m_i = \mathcal{N}_1$$

**Environment scenario CD[B]**

The number of reals is uncountably infinite on an unbounded line:
\[ X = \mathcal{R}, \quad |X| = \mathcal{N}_i \]  

Since there is nothing here to vary with time, for this environmental scenario:

\[ m_i = m_i = \mathcal{N}_i \]

**Environment scenario C[D]B**

As defined in Chapter 5, noise maps a continuum of points to distinguishable intervals as defined by the standard error width \( w \). From our previous analysis, we will consider \( w \) as a discretization of the input space. Thinking in one dimension for simplicity, for an environment of length \( l_o \), the number of inputs is:

\[ |X| = l_o / w \]

If length is finite (if the universe is bounded), and \( w \) is (instantaneously) finite, then this will produce a finite number of states:

\[ m_i = |X| = l_o / w = \mathcal{N}_{\text{max}} \]

However, \( w \) varies with time:

As \( \Delta t \to \infty, \quad w \to 0 \)

Since \( w \) can be arbitrarily small in the limit, the number of inputs becomes countably infinite:

\[ m_i = |X| = l_o / w = \mathcal{N}_0 \quad \text{as} \quad \Delta t \to \infty \]

**Environment scenario C[D][B]**

From the argument related to scenario \( E; \ C[D]B \) above, if length is infinite
then the number of states will be infinite. However, because the noise has
discretized the number line, the number of states must be countable:

$$|X| = \frac{l_\theta}{w} = \mathcal{N}_0$$

Though, again, $w$ varies with time, arbitrarily small intervals (in the limit) on an
unbounded line are still countable. Thus:

$$m_i = m_t = |X| = \frac{l_\theta}{w} = \mathcal{N}_0$$

**Environment scenario [C]DB**

By definition, if space-time is not continuous (discrete), then there is a
minimum chunk size, $d$, analogous to $w$ (though $d$ does not vary with time). We
can, therefore, use the same arguments as above

$$|X| = \frac{l_\theta}{d}$$

If physics is deterministic, then no noise is introduced:

$$|X| = \frac{l_\theta}{d}$$

If length is finite, then this will produce a finite number of states:

$$|X| = \frac{l_\theta}{d} = \mathcal{N}_{\text{MAX}}$$

Since $d$ does not vary with time:

$$m_i = m_t = |X| = \frac{l_\theta}{d} = \mathcal{N}_{\text{MAX}}$$

**Environment scenario [C]D[B]**

Argument is as for [C]D[B]. If length is infinite then the number of states will
be infinite. However, because the state space is discretized, the number of
states must be countable:

\[|X| = \frac{l_\Theta}{d} = \aleph_0\]  

\[\Theta [C][D][B]\]

Since \(d\) does not vary with time:

\[m_i = m_t = |X| = \frac{l_\Theta}{d} = \aleph_0\]  

\[\Theta [C][D][B]\]

**Environment scenario [C][D][B]**

Using the results from \(C[D]\) and \([C]\), we can consider the case of a non-deterministic physics with discrete space-time as follows.

\[|X| = \frac{l_\Theta}{wd}\]  

\[\Theta [C][D]\]

Here, because \(d\) is the minimum physically-realizable volume of state space, it cannot be reduced by \(w\), only increased. Thus \(w\) is constrained to be at least 1. It can also be argued that, in this case \(w\) must be an integer. However, as this does not make any difference to our argument we will not try to restrict \(w\) further. For a bounded universe, this leads to a decrease in the potential number of states:

\[|X| = \frac{l_\Theta}{wd} = N_{\text{MAX}}\]  

\[\Theta [C][D][B]\]

\[m_i = |X| = \frac{l_\Theta}{wd} < \frac{l_\Theta}{d} = N_{\text{MAX}}\]  

\[\Theta [C][D][B]\]

Because of the existence of \(w\), there is a difference over time

\[m_i = |X| \rightarrow \frac{l_\Theta}{d} = N_{\text{MAX}}\]  

\[\Theta [C][D][B]\]

**Environment scenario [C][D][B]**

Again, because there is a minimum chunk size, we know the number of states
in the unbounded space must be countable. This is the case both instantaneously and over time:
\[ m_i = m_t = \mid X \mid = l_\Theta/wd = \aleph_0 \]

\[ \Theta \text{[C][D][B]} \]

### 6.2.2.2 Possible scenarios for the system

As we defined them in Chapter 4, the system and environment are identical except that the former is constrained to be bounded and the latter is not, and the former is constrained to be larger than or equal to the latter. Thus, the bounded environmental scenarios can be used for the system. The only difference is that we must substitute \( l_z \) for \( l_\Theta \). For completeness, these results are:

\[ n = \mid X^* \mid = \aleph_1 \]

\[ \Sigma \text{CDB} \]

\[ n = \mid X^* \mid = l_z/w = \aleph_{\text{MAX}} \]

\[ \Sigma \text{C[D]B} \]

\[ n = \mid X^* \mid = l_z/d = \aleph_{\text{MAX}} \]

\[ \Sigma \text{[C]DB} \]

\[ n = \mid X^* \mid = l_z/wd = \aleph_{\text{MAX}} \]

\[ \Sigma \text{[C][D]B} \]

### 6.2.2.2 Physical information retention and specific virtual machines

Now we can consider the physical information retention for the various physical scenarios—using the calculation made using the method described in Section 6.2.1—and consider how they affect the power of specific types of computer defined in Chapter 5.
We will go through the three different types of machine—analogue, Turing, and strictly-precision-limited digital computer—in turn. Note that whether the analogue machine is noisy or not is inherent in the physical conditions in which it exists, therefore noisy and perfect analogue systems need not be treated separately.

Because we are interested in machines that can be physically implemented in a given scenario, we will, in each case, consider that the number of inputs for each is constrained to be no better than the maximum possible for $\sum$ for the given physical scenario defined in the last section. This is the maximum that can be performed by a physical computer. Therefore, where the maximum number of inputs in the virtual machine and physical system are different, we will determine information loss using the lower of the two cardinalities. The results are shown in Table 6.1 below.

**Table 6.1:** Information retention for machines implemented on physical computers in different environmental scenarios. Note both virtual and real interaction cases are included below.

<table>
<thead>
<tr>
<th>Option</th>
<th>Machine</th>
<th>C</th>
<th>D</th>
<th>B</th>
<th>Instantaneous physical information retention</th>
<th>Physical information retention over time</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1.1</td>
<td>Analogue</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>$(N_1, N_1) = 1$</td>
<td>$(N_1, N_1) = 1$</td>
</tr>
<tr>
<td></td>
<td>Turing</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>$(N_0^r, N_1) = 0$</td>
<td>$(N_0^r, N_1) = 0$</td>
</tr>
<tr>
<td></td>
<td>SLPDC</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>$(\nabla_{max}, N_1) = 0$</td>
<td>$(\nabla_{max}, N_1) = 0$</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Analogue</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>$(N_1, N_1) = 1$</td>
<td>$(N_1, N_1) = 1$</td>
</tr>
<tr>
<td></td>
<td>Turing</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>$(N_0^r, N_1) = 0$</td>
<td>$(N_0^r, N_1) = 0$</td>
</tr>
<tr>
<td></td>
<td>SLPDC</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>$(\nabla_{max}, N_1) = 0$</td>
<td>$(\nabla_{max}, N_1) = 0$</td>
</tr>
<tr>
<td>Option</td>
<td>Machine</td>
<td>C</td>
<td>D</td>
<td>B</td>
<td>Instantaneous physical information retention $(\mathcal{X}, \mathcal{X}) = P_i$</td>
<td>Physical information retention over time $(\mathcal{X}, \mathcal{X}) = P_i$</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Analogue</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>$(\frac{l_x}{w}, \frac{l_\alpha}{w}) = \frac{l_x}{l_\alpha}$</td>
<td>$(\frac{l_x}{w}, \frac{l_\alpha}{w}) = \frac{l_x}{l_\alpha}$</td>
</tr>
<tr>
<td></td>
<td>Turing</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>$(\frac{l_x}{w}, \frac{l_\alpha}{w}) = \frac{l_x}{l_\alpha}$</td>
<td>$(\frac{l_x}{w}, \frac{l_\alpha}{w}) = \frac{l_x}{l_\alpha}$</td>
</tr>
<tr>
<td></td>
<td>SLPDC</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>$(\mathcal{N}<em>{\text{max}}^*, \frac{l</em>\alpha}{w}) = \mathcal{N}<em>{\text{max}}^* \frac{l</em>\alpha}{w}$</td>
<td>$(\mathcal{N}<em>{\text{max}}^*, \frac{l</em>\alpha}{w}) \rightarrow 0$ as $\Delta t \rightarrow \infty$</td>
</tr>
<tr>
<td>6.1.4</td>
<td>Analogue</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>$(\frac{l_x}{w}, \mathcal{N}_0) = 0$</td>
<td>$(\frac{l_x}{w}, \mathcal{N}_0) \rightarrow 1$ as $\Delta t \rightarrow \infty$</td>
</tr>
<tr>
<td></td>
<td>Turing</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>$(\frac{l_x}{w}, \mathcal{N}_0) = 0$</td>
<td>$(\frac{l_x}{w}, \mathcal{N}_0) \rightarrow 1$ as $\Delta t \rightarrow \infty$</td>
</tr>
<tr>
<td></td>
<td>SLPDC</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>$(\mathcal{N}_{\text{max}}^*, \mathcal{N}_0) = 0$</td>
<td>$(\mathcal{N}_{\text{max}}^*, \mathcal{N}_0) = 0$</td>
</tr>
<tr>
<td>6.1.5</td>
<td>Analogue</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$(\frac{l_x}{d}, \frac{l_\alpha}{d}) = \frac{l_x}{l_\alpha}$</td>
<td>$(\frac{l_x}{d}, \frac{l_\alpha}{d}) = \frac{l_x}{l_\alpha}$</td>
</tr>
<tr>
<td></td>
<td>Turing</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$(\frac{l_x}{d}, \frac{l_\alpha}{d}) = \frac{l_x}{l_\alpha}$</td>
<td>$(\frac{l_x}{d}, \frac{l_\alpha}{d}) = \frac{l_x}{l_\alpha}$</td>
</tr>
<tr>
<td></td>
<td>SLPDC</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>$(\mathcal{N}<em>{\text{max}}^*, \frac{l</em>\alpha}{d}) = \mathcal{N}<em>{\text{max}}^* \frac{l</em>\alpha}{d}$</td>
<td>$(\mathcal{N}<em>{\text{max}}^*, \frac{l</em>\alpha}{d}) = \mathcal{N}<em>{\text{max}}^* \frac{l</em>\alpha}{d}$</td>
</tr>
<tr>
<td>6.1.6</td>
<td>Analogue</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>$(\frac{l_x}{wcd}, \frac{l_\alpha}{wcd}) = \frac{l_x}{l_\alpha}$</td>
<td>$(\frac{l_x}{wcd}, \frac{l_\alpha}{wcd}) = \frac{l_x}{l_\alpha}$</td>
</tr>
<tr>
<td></td>
<td>Turing</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>$(\frac{l_x}{wcd}, \frac{l_\alpha}{wcd}) = \frac{l_x}{l_\alpha}$</td>
<td>$(\frac{l_x}{wcd}, \frac{l_\alpha}{wcd}) = \frac{l_x}{l_\alpha}$</td>
</tr>
<tr>
<td></td>
<td>SLPDC</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>$(\mathcal{N}<em>{\text{max}}^*, \frac{l</em>\alpha}{wcd}) = \mathcal{N}<em>{\text{max}}^* \frac{l</em>\alpha}{wcd}$</td>
<td>$(\mathcal{N}<em>{\text{max}}^*, \frac{l</em>\alpha}{wcd}) = \mathcal{N}<em>{\text{max}}^* \frac{l</em>\alpha}{wcd}$</td>
</tr>
<tr>
<td>6.1.7</td>
<td>Analogue</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>$(\frac{l_x}{d}, \mathcal{N}_0) = 0$</td>
<td>$(\frac{l_x}{d}, \mathcal{N}_0) = 0$</td>
</tr>
</tbody>
</table>
Certain properties of these results should be noted. First, the value of $P$ for information retention in non-deterministic scenarios increases with $t$, only if the actual noise distributions have properties similar to those discussed in Chapter 5. This assumption is supported by the central limit theorem but is not necessarily supported by physics. This issue will be discussed further in Chapter 7.

Second, consider those scenarios where the environment is discretized: either through intrinsic noise or through lack of continuity (Option 6.1.3 and all that follow). As would be expected, in all these cases, analogue and Turing machines have identical values for information retention. This is as we would expect, because any continuity advantage of the analogue system is lost. Likewise, it is not surprising that, where the continuum does exist and is not discretized by noise, there is a clear analogue advantage.

What comes out is that, in every single environmental scenario, the

<table>
<thead>
<tr>
<th>Option</th>
<th>Machine</th>
<th>C</th>
<th>D</th>
<th>B</th>
<th>Instantaneous physical information retention $(\mathcal{X}, \mathcal{X}) = P_i$</th>
<th>Physical information retention over time $(\mathcal{X}', \mathcal{X}) = P_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turing</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>$(\frac{I}{\mathcal{d}}, \mathcal{N}_0) = 0$</td>
<td>$(\frac{I}{\mathcal{d}}, \mathcal{N}_0) = 0$</td>
</tr>
<tr>
<td>SLPDC</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>$(\mathcal{N}^{\text{max}}, \mathcal{N}_0) = 0$</td>
<td>$(\mathcal{N}^{\text{max}}, \mathcal{N}_0) = 0$</td>
</tr>
<tr>
<td>6.1.8</td>
<td>Analogue</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>$(\frac{I}{w_d}, \mathcal{N}_0) = 0$</td>
<td>$(\frac{I}{w_d}, \mathcal{N}_0) = 0$</td>
</tr>
<tr>
<td></td>
<td>Turing</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>$(\frac{I}{w_d}, \mathcal{N}_0) = 0$</td>
<td>$(\frac{I}{w_d}, \mathcal{N}_0) = 0$</td>
</tr>
<tr>
<td></td>
<td>SLPDC</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>$(\mathcal{N}^{\text{max}}, \mathcal{N}_0) = 0$</td>
<td>$(\mathcal{N}^{\text{max}}, \mathcal{N}_0) = 0$</td>
</tr>
</tbody>
</table>
analogue machine has as high a physical information retention as the Turing machine or better: this is true whether the noise can be minimized or not. Likewise, the analogue machine is always as good as or better than (usually considerably better than) the strictly-limited-precision digital computer. This is not surprising: the Turing machine is forced to run on the physical (analogue) machine, so it cannot do better than it.

Nevertheless, in scenarios 6.1.7 and 6.1.8, all machines retain the same proportion of information. Given the ease of building the SLPDC, should these scenarios apply, then the digital computer has a clear technological advantage without any theoretical disadvantage.

5.3.3. Reality check using robot example

If we consider our wheeled robot from Chapter 4, the results in Table 6.1 make sense. Classically, in mathematically-continuous space-time, the ever-changing state of the universe could easily be unambiguously represented: the rotational positions of one of the wheels alone could encode the information. The number of states in the former, uncountably infinite, could theoretically be exactly mapped onto the number of states of the latter. However, as soon as discretization takes place (only certain rotations allowed, for instance), the number of states of the wheel becomes negligible in comparison to those in the universe because of the mismatch in size. In between these two extremes, some proportion of the information about the larger system should be reflected in that of the smaller.
Question 6.1.4 Can virtual interactions approximate real interaction?

Where virtual machines lose information ($P \neq 1$), they may nevertheless be able to approximate the behaviour of physical systems. In particular, we are interested in approximation for physical scenarios $\text{CD}B$ and $\text{CD}[B]$, where:

\[ X = \mathbb{R}, |X| = \aleph_1 \]

\[ X' = \mathbb{R}, |X'| = \aleph_1, \] that is, where real interaction, in principle, can take place.

We can prove that approximation to arbitrary precision is possible for Turing machines in the following way.

6.1.4.1 The approximation hypothesis

Assumption 1: Every real number can be approximated by a rational number to arbitrary precision: for all $x \in \mathbb{R}$ and $\delta x > 0$ there exist $n, m \in \mathbb{N}$ such that:

\[ |x - n/m| < \delta x \]

Assumption 2: $G$ is continuous: for all $\epsilon_2$ there exists $\delta x_2(\epsilon_2)$ such that $|x - a| < \delta x_2(\epsilon_2) \Rightarrow |G(x) - G(a)| < \epsilon_2$

Assumption 3: $G(x') = \text{def} V(x')$

Proposition: For every $\epsilon$ there exists a function $F_{in}: \mathbb{R} \rightarrow \mathbb{Q}$ such that for all real $x \in \mathbb{R}$ there exists a $\delta x$ such that $x \in \mathbb{R} \rightarrow |G(x) - V F_{in}(x)| < \epsilon$
This can be restated as follows: every real interaction can be approximated by a virtual Turing function to arbitrary precision.

Proof: Assume $\varepsilon$ is given. Let $\varepsilon=\varepsilon_2$. From Assumption 2, there exists $\delta x_2$ such that $|x-a|<\delta x_2 \Rightarrow |G(x)-G(a)|<\varepsilon_2$. Let $\delta x_1=\delta x_2$. From Assumption 1, for all $x \in \mathbb{R}$ there exist $n, m$ such that:

$$\left|x - \frac{n_\beta}{m_\beta}\right| < \delta_1$$

Let $F_{in}(x) = \frac{n_\beta}{m_\beta}$

Then $|x - F_{in}(x)| < \delta x_1 \Rightarrow |G(x) - VF_{in}(x)| < \varepsilon_2$

Though proven here for $Q$, this can also be proven for $C$ (which is, in any case, defined as a superset of $Q$). It can also be proven for $D$ by rewriting Assumption 1 as follows.

Assumption 1.1: Every real number can be approximated by a distinguishable value to arbitrary precision given sufficient time: for all $x \in \mathbb{R}$ and $\delta x_1 > 0$ there exists a $w_i$ such that $w_i < \delta x_1$.

Note that the hypothesis only applies where the function being approximated is continuous.

6.1.4.2 Finding a sufficient approximation

Let us consider how to find the appropriate $\delta x$ for a given system in the following conditions:
1. $G_t$ is known for all time

2. $G_t$ is unknown but constrained to a finite set

3. $G_t$ is unknown and a member of an infinite set

**Condition 1:** $G_t$ is known for all time

Consider a virtual machine is being designed to approximate a specific physical system, with $G_t$ known for all time. Since $\partial x$ is dependent on $G_t$ for a given $\varepsilon$, then $\partial x$ must also be a function of time. In this case, we can simply plot this function (call it the error function, $\delta$) and choose any $\partial x \leq \delta_{(\min)}$

**Condition 2:** $G_t$ is unknown but constrained to a finite set

Consider that a designer does not know how $G_t$ will evolve, but does knows every possible $G \in G$ because $G$ is a known, finite set. In this case, for a given $\varepsilon$, we can calculate $\partial x_1, 2, 3, \ldots$ for each $G_1, 2, 3, \ldots$. We can then choose $\partial x$ such that the smallest of all $\partial x_1, 2, 3, \ldots > \partial x$.

**Condition 3:** $G_t$ is unknown and a member of an infinite set

Consider now Condition 2 above, but $G \in G$, where $|G| = \aleph_0$. Consider that $G$ consists of functions $y = mx$, where $m$ is the gradient of the line. Also, for simplicity, let us consider positive $\partial x$ only. Here, for a given $\varepsilon$, we have:

$$\varepsilon = |m(x + \partial x) - y|$$

$$\partial x = |(y + \varepsilon)/m - x|$$

Now, for any value of $\partial x$ that we choose, we can also find a value of $m$ such that the approximation condition is not met. Thus, if $\partial x$ must be set before we
have specific knowledge of $G$, it is not possible to confirm that the approximation will be within our error margins. On the other hand, if $\partial x$ can be specified as needed (as in the noisy analogue case) then a $\partial x$ meeting our specified approximation can be found if given enough time.

**Conclusions from this chapter**

- Both the machine type (for environment and system) and interaction type are dependent on the answers to the following physical questions:
  
a. The continuity or not of space-time and other physical dimensions continuous?

  b. Is physics deterministic?

  c. Is the universe bounded?

- Analogue and Turing machines, and strictly-limited-precision digital computers may have different retention of physical information whilst being in the same (virtual) interaction class. The actual capacity is determined by issues such as the physical scenario, the minimum sizes of any discretizing element (whether noise or actual discreteness), and the relative sizes of the object and environment. In the analogue and Turing cases, the question of whether or not the noise they experience is inferential is also a factor. Analogue information retention is always as good as, or better than, that of Turing machines, which is always as good as or better than SLPDCs.

- Noisy analogue and Turing virtual interactions may approximate real
interaction to a given precision in conditions where the embodied agent's function \((G)\) is both known as it varies in time and continuous. If the former condition is not met, only analogue systems may be able to perform the approximation. If \(G\) is not continuous then neither machine may perform the approximation.

**Issues raised in this chapter to be discussed further**

- The validity of the various physical options and the noise assumption will be considered in Chapter 7.

- The engineering implications of the various options and the problems with functions and approximation will be considered in Chapter 8.

**Assumptions and discussions beyond the scope of this thesis**

- In order to present a general view, the complexity (or lack of it) in the embodied intelligent agent and environment is not considered here: we only consider the interactions between them.
Chapter 7:

The interface with physics

Based on the preceding arguments and analysis, it should be clear that both the type of interaction that a machine has with its environment and the information that it is able to retain (or not) are determined by physical questions. In this chapter, we will consider whether physics has answers to these questions. In particular, these are:

1. Is space-time continuous or discrete? (C or [C]?)
2. Is the universe deterministic? (D or [D]?)
   2b. Can the effects of noise be minimized over time \(m \neq m_t\)?
3. Is the universe bounded or not? (B or [B]?)

Note that issues related to noise are connected with the issue of determinism, as will be discussed.

It is also important to note at this stage that all of these questions are somewhat difficult and controversial issues in physics. The intention here is not to answer the questions outright (there are no agreed-upon answers at this time) but to point to the areas of physics from which answers might eventually come. More importantly, in this section we hope to show that questions that might appear to those outside physics to be already decided this particularly relates to the issues of discreteness and determinism are not clear cut. In fact, these are important and active areas of study in physics.
Chapter 7: The interface with physics

**Question 7.1: Is space-time continuous or discrete (C or [C])?**

This is a complicated question and answers to it lie on the frontiers of physics research. The Newtonian view of space-time, the basis of classical mechanics (Kibble, 1985), was that it was both absolute and continuous. The former view was changed by Einsteins' special (and later, general) relativity (Einstein & Infeld, 1967). The absoluteness of space-time is not at issue here, so we will not discuss it further, though the conflict with relativity will become important later. The second property of Newtonian space-time, continuity, may be at odds with quantum mechanics.

However, this is a more complex question than is normally understood outside the immediate field. In conversations with scientists and engineers without a physics background, a common misconception has become apparent. Many believe that quantum mechanics says that the universe is discrete, not continuous, as evidenced by the existence of the discrete energy levels in atoms. The discreteness of packets of energy and matter electrons, photons, etc. appear to back up this view. Both of these questions relate to the issue of quantization, which can mean two things. The first possibility, based on established physics and discussed, has to do with the issue of quantization of light and the discrete number of states available to finite systems. The second, based on some theories of quantum gravity, is that the space-time itself is inherently quantized.
7.1.1 Quantum-mechanical discreteness

In conventional quantum mechanics (QM), quantization refers to the fact that certain properties of matter at the microscopic level are constrained to vary discretely. For instance, the energy of a photon with frequency $\nu$ is constrained to be an integral multiple of $\varepsilon$, the minimum energy, where $\varepsilon = h \nu$. The $h$ is Planck’s constant, which figures heavily throughout quantum mechanics. Another well-known example of this phenomenon concerns the orbits of electrons in atoms: in the simplest case, their angular momenta (electron mass $m$ times its velocity $v$ times radius of orbit $r$) are restricted to take values of $n\hbar/2\pi$. Because of the discretization of orbits (often known as energy levels), an atom is considered to be in one of a number of discrete quantum states based on the activities of its surrounding electrons. (For a full derivation of these results, and a mathematical treatment of the basics of quantum mechanics, see, for example, (Gasiorowicz, 1974).)

Quantization should not be confused with discreteness in the computational sense. For a particular frequency, energy is discretized. However, $\nu$ can vary continuously, so the number of unique photon energies that could be created, given the right conditions, also varies continuously (assuming that space-time is otherwise continuous, discussed in the next section). Likewise, though the angular momentum of electrons is constrained as has been discussed, $r$ may vary continuously. This latter situation is a particularly interesting case, and the basis of semiconductor engineering. The
energy levels available to an electron change when, for instance, an electromagnetic field is introduced to force the particle in a particular direction. (For an intuitive understanding of quantum mechanics and its applications, we highly recommend (Hey & Walters, 1987).)

In other words, though constrained to take particular configurations given their particular circumstances, quantum systems may still vary continuously with appropriately varying stimuli.

7.1.2 Quantum gravity, string theory, and discrete space-time

Though quantum mechanics is an extremely reliable predictive theory, it is not seen by physicists as a final complete theory. Essentially, it describes how things are without explaining how they work. As is generally understood in the physics community (see e.g. (Smolin, 2000) and (Greene, 2000)), the main problem is that quantum mechanics and general relativity—which has proved to provide just as excellent predictions about large-scale and astronomical phenomena as QM has about sub-atomic interactions—are incompatible. This fact can be seen particularly well when considering two cases, black holes and the big bang, because these are both circumstances where bodies are extremely dense (high in mass, small in volume).

Clearly, it is not appropriate to go into the rationale or details of grand unification theories (theories that unify QM and gravity or general relativity) here. However, it is equally clear that this is a major mainstream effort in physics today, and that ignoring it in favour of non-explanatory theories like
quantum mechanics would not fulfil the goals of this chapter. Therefore, it is important to take into account of current approaches and where they might lead, even if they are as yet inconclusive. If all point to the same outcome in terms of our interest here—spatial continuity or not—then we can take that as a strong indication that this is the right answer.

To start off, we should point out that, space is definitely continuous under all the basic physics (including QM) we use today: all of the mathematical constructs used to design the evolution of the wave equation, etc., are continuous. However, the issue of potential discreteness arises because there is a length (known as Planck's length, approximately $10^{-35}$ m: see (Anderson, 1989)) at which physics as we know it breaks down. There are at least three different approaches to solving this problem (as Smolin explains in his book) but we will consider only two here: lattice based approaches to loop quantum gravity (Smolin's preferred road) and string theory unified by M-theory (Greene's favoured option).

Without going into detail, Smolin's view (which represents a significant branch of the theoretical physics community) is that the very concept of space-time is meaningless. This school of thought is that space only exists in terms of relationships between objects in the universe. One way to understand this is to imagine the universe exploding in a big bang. Is it meaningful to consider the space outside the expanding bubble of matter as part of the universe? The loops in the name of the theory he espouses represent these relationships: one way to think of them is as interaction pathways, each one binding each
particle to every other particle.

The view of space-time that he believes is most appropriate to this approach is that it is both relative and lattice-based. In other words, the lattice is not fixed to some external view of the universe, but integral to the particular relationship being considered. In this scenario, the lattice has points that are of the order of a Planck's length apart. As a result it appears to be meaningless to consider distances that are less than an integral number of Planck lengths long. Essentially, \( n + 1/2 \) Planck lengths would have to be considered to be the same as \( n \) or \( n + 1 \) Planck lengths. Though Smolin does not go into the geometrical arguments, presumably the relative nature of the lattice would imply that one axis was always in the direction connection the two objects (thus avoiding the problem that going 1 Planck length up and 1 across would involve a distance of \( \sqrt{2} \)). Alternatively, some non-Euclidean space could be involved. Though the details are sketchy, and the concept of relative space and time is hard to map into our model, such a theory—if proved correct—would most likely be considered a fully discrete scenario.

Greene's favoured theory, currently known as M-theory (for mystery, magical or matrix, according to Ed Witten who coined the name) is in fact a way of joining together five existing string (short for superstring) theories through the addition of a spatial dimension. However, the string theories under discussion already have ten dimensions: time, the three well-understood spatial dimensions, and a further six curled-up dimensions. Green likens the
difference between an ordinary and curled up dimension to the difference between the length and circumference of a telegraph wire—from the perspective of an ant walking on the wire—respectively. The addition of the 11th dimension allows different, but similar, ten-dimensional theories to be (potentially) unified as being different aspects of the same thing.

The issue of the curled up dimensions will be discussed further in considering the issue of determinism, but is not particularly crucial to our question of continuity. What is important is that, in string and M theories, space is still assumed to be continuous. The issue of strange behaviour at distances below Planck's length is looked at in very different terms. According to Greene, as strings shrink below the Planck length the calculated impact on the interaction with other objects is as if the string had reached the Planck length and then started growing again. It's not clear exactly what this means: whether there is a minimum volume below which matter/energy cannot be trapped, or whether it simply means that there is an ambiguity in whether a string is a little smaller or a little larger than this threshold. However, there is no clear suggestion that this minimum size provides a means for discontinuity. As long as strings are larger, it appears (so far) that they can take any size.

According to Smolin, quantum gravity—or, indeed, any grand unified theory—will require an unequivocally discrete universe. Greene, on the other hand makes no such claim. So, in fact, this issue is far from settled even theoretically. Further, even once the theoreticians have come to some kind of consensus, there is the issue of experimental proof that any particular theory is
actually correct. Currently, even potential sources of experimental verification are purely speculative. Indeed, one of the chief astronomical objects that has been the inspiration for thought experiments and thinking about the quantum gravity issue—the black hole—has not yet been proven positively to exist (thought there is compelling evidence that they do). Further, our understanding of the nature of black holes—especially in terms of information—has changed dramatically even in recent months (Rodgers, 2004). Finally, even once the theory has been both agreed on and verified, it may take some time to come to agreement on what it actually means. The interpretation of physical theories, even those with tried and tested predictive power such as quantum mechanics, can lie unresolved for some time (in much the same way as has happened in quantum mechanics, see e.g. (Squires, 1994) and the discussion to answer question 7.2).

**Answer to Question 7.1: Is space-time continuous (C or [C])?**

From what we have learned about quantum gravity and string theory, we know that this question is still not decided. Based on current knowledge either the C or [C] scenarios may be valid.

**Question 7.2: Is the universe deterministic (D or [D])? Can the effect of noise be minimized with time?**

These questions must, initially, be considered together. At their root is the nature of physical noise, a subject that is commonly misunderstood in
engineering contexts. In the first two of the following sections we will discuss this issue in detail before attempting to tackle the other questions head on.

7.2.1 Is physics an information theory?

Most analyses related to real computing machinery (hardware) are based on Shannon information. As described fully in Chapter 5, Shannon information is based on the premise that, in an interaction, a message is being sent from a sender to a receiver via a noisy channel. It was Shannon who showed that—with normally-distributed noise and finite bandwidth—only a discrete number of distinguishable states could be sent using an analogue communication channel. This means that the analogue signal can be represented discretely without loss of information (defined as the number of distinguishable states). As well as transmission lines, devices such as digital sensors, actuators, and displays are designed based on Shannon information. For instance, cameras and computer screens are built knowing that the eye can only distinguish between light levels with a particular minimum difference in luminance: designers need only push the technology to that point. Similarly for digital audio recordings.

In the literature, models of computation and artificial intelligence are based on this sender-channel-receiver model. For example, in a generally philosophical consideration of issues of the issue of the emergence of mind from matter, Mulhauser (Mulhauser, 1998) raised the issue of an absolute bound on computer speed determined by Bekenstein (Bekenstein, 1981) as an
argument that the brain must have a similar bound. Bekenstein's analysis, though based on thermodynamics and causality considerations, sought to find the maximum rate at which Shannon information could be transferred.

Fisher information is another way of thinking about the concept of sending and receiving: more specifically, the concept of measurement (Matthews, 1999). It allows us to understand how much information can be gleaned using a particular apparatus, and by understanding this quantity, to maximize it. One example of an engineering use for Fisher information is wavelength coding (Cathey & Dowski, 2002), which allows the amount of information captured by an imaging system to be maximized for a given application. Though the amount of information acquired remains constant, what information is acquired is not. This technique has been used to extend depth of focus in imaging systems. The main point to note here is, simply, that Fisher-information-based analyses (see, e.g. (Frieden & Soffer, 2000)) are similar to their Shannon-information counterparts in that they relate to the measurement of specific and defined features of the environment, not a general physical interaction.

For this reason, concepts of Shannon and Fisher information do not apply to the real interaction case in our model: it is based on physics, not knowledge. As we have defined it, the embodied intelligent agent need not 'know' where the various influences upon it are coming from. In fact these influences are entirely ambiguous, leaving it free to act (or, in fact, constrained
to act) as dictated by their cumulative physical influence. An information-based device, on the other hand, is built in such a way that the 'sensing the universe' and 'acting' functions within it are de-coupled: virtual interaction. This provides an extra degree of freedom, making the system very easy to design. But it necessitates an interface between the physical and computational, or, in a sense, between the real and virtual worlds. This interface is information.

This trade-off of knowledge against action is actually a very deep philosophical point in physics and one that is often poorly understood. This poor understanding is a result of confusing the mechanism of physics itself with the mechanism through which we understand physics. The Heisenberg uncertainty principle (Heisenberg, 1930) is an excellent example of this. The uncertainty principle says that by measuring (for instance) the position of a particle, you lose the ability to acquire information about its momentum. This is not a principle of physics but one of measurement and knowledge: it does not necessarily mean that a particle does not have position and momentum at the same time (although this may be true for other reasons), but rather that the two cannot be measured and known simultaneously with accuracy. This is important because it helps us to design meaningful experiments and to interpret the results correctly. It does not, however, give us answers about what the particle would do if left to its own devices: unknown and unmeasured by us.
7.2.2 What is noise?

Understanding the difference between real physical interaction and both measurement and communication is important because it allows us to more tightly define what noise is and what deterministic means. When physicists talk about noise they mean one of the following:

a. They built a theoretical model of a phenomenon or experiment, and their model did not take account of all the possible influences at work: thus the result of the experiment and model differed. (We will refer to this as modelling noise)

b. They built a theoretical model that specifically ignored microscopic activity (this is the basis of statistical mechanics and thermodynamics, for instance). When the experiment was performed, the measurement did not take place over a long enough period to accommodate this statistical basis (this may be necessary, and not bad experimental design), thus the experimental results did not agree with the theoretical results. (Statistical noise)

c. They built their theoretical model, which was correct, but they were unable to measure the initial conditions or final results accurately enough for perfect agreement. (Measurement noise)

d. They built their theoretical model but something unexpected happened during the experiment (like dropping equipment, a power surge) to cause either a noise or other disturbing signal. (Traditional noise)

e. Quantum-mechanical non-determinism affected their measurements.
(Quantum noise)

Note that, apart from quantum noise—which we will define as including phenomena such as zero-point fluctuations—all of these definitions require both a model and a measurement for the noise to become apparent: it lies in the mismatch between the two. This is similar to Shannon information theory, where the noise becomes apparent when looking at the mismatch between the sent and received message. So, the noise isn't inherent in the physics, but rather in our modelling and measurement of the physics. For physical systems that experience real interaction, therefore, only quantum noise is meaningful.

### 7.2.3 What is quantum noise?

Among physicists, the nature and meaning of quantum noise is still an extremely controversial issue. The problem falls out of the mathematics of quantum mechanics (see, e.g. (Gasiorowicz, 1974)) and can be summed up as follows. Using Schrödinger's wave equation and associated mathematics we can very accurately predict what will happen next in quantum systems. However, we can only do this statistically—for large groups of systems—not for individuals. Thus, all the predictions we make are limited to probabilities. For instance, we might know that half the photons hitting a half-silvered mirror will go in one direction and half in the other, but we do not know in which direction a particular photon will go. The spontaneous emission of a photon from an atom, for instance, could happen at any time (though there will be a higher likelihood that it happens at one time than another). This is the reason
why radioactive samples are discussed in terms of half-lives, a statistical
measure, rather than some more precise formula.

The source of the probabilistic nature of quantum mechanics (the fact of
its existence is not controversial) is a topic of continuing debate and research.
Rather than considering the possible alternatives in detail, which would be
inappropriate here, we will concentrate solely on how different interpretations
support different environmental scenarios: in this case, deterministic and non-
deterministic physics. Note that what follows are just interpretations of an
established theory. Since (so far) they cannot be told apart by experiment, they
are beliefs and not facts. Also, this research is still active, so absolute
conclusions are not possible.

Interpretations of quantum mechanics generally fall into two categories:
deterministic and non-deterministic. In the former case, the result of a
quantum interaction is considered to come from some physical factor that is as
yet (or in principle) unknown to us. Thus, though the outcome may appear to
be probabilistic, there is a causal chain of events leading up to it. Interpretations with this underlying belief are generally known as hidden-
variable theories.

The term hidden variable can be thought of in two different ways. First, it
refers to some very specific theories about how the apparently probabilistic
nature of quantum mechanics can emerge from laws of physics that are
deterministic. Second, it refers to the broad class of theories (including those
yet to be conceived or invented) that address this issue: in general we will use
the latter meaning. Conceptually, hidden variables emerge from the fact that, when performing quantum-mechanical experiments, carefully-controlled events that appear to be identical nevertheless have different outcomes. However, when repeated a sufficient number of times, the results of these experiments will tend towards a distribution that can be predicted by (probabilistic) quantum theory. The lack of determinism in the individual case, combined with a seemingly deterministic collective behaviour, is a very basic difference between quantum theory and classical physics and was both philosophically and conceptually difficult for physicists to accept. Hidden variables side-step this issue by asserting that, though they appeared to be identical, the experiments, in fact, are different. These differences are embodied in the hidden variables.

The hidden variables in question cannot be related to the particles being measured alone: they are non-local. This fact came out of the experimental demonstration of a famous thought experiment (the Einstein-Podolsky-Rosen or EPR (Einstein et al. 1935) experiment), and the subsequent development of the Bell inequalities (Bell, 1966). Essentially this means that, if there is a hidden variable determining quantum-mechanical behaviour, then it must apply to the entire quantum mechanical system, and not just to an individual element of it.

Hidden variables theories vary widely, and this is not an appropriate place to discuss them. Some, like Bohm's theory (Bohm, 1952), have been
largely discredited, but work continues in the field (see e.g. ('t Hooft, 2001; Coecke, 2002; Valentini, 2002)).

It is important to note that these theories are deterministic in the sense that the outcome is caused by a chain of events with no need of a random variable. However, they are still practically non-deterministic (which could be because they are chaotic), in the sense that we cannot, even in principle, know what the outcome will be in advance. This is thanks to Heisenberg’s uncertainty principle (see e.g. (Gasiorowicz, 1974)), which states that it is not possible to know with unlimited accuracy both the position and momentum of a particle. Essentially, this means that we can never have enough information to make an accurate global prediction, regardless of whether some of the variables are hidden or not. The results of this class of theories are therefore identical in their predictive value to their non-deterministic counterparts.

Note also, that actual rather than practical non-determinism is required in order for it to effect our model. Otherwise, quantum noise like the other forms we discussed would be caused by our inability to measure or model: neither of which our intelligent system must do.

The best known of the non-deterministic theories is the so-called many-worlds interpretation (Squires, 1994). In fact, it is deterministic in the sense that we know exactly what will happen: all of the quantum possibilities occur every time a quantum event takes place, each possibility spawning its own separate universe. From the perspective of our model, which only deals with the physics
of a single universe, the theory is effectively non-deterministic because we do not know which path we will be taking through this expanding tree of universes.

The Copenhagen interpretation (Gasiorowicz, 1974), which was taught to physicists for much of the latter 20th century, is, in a sense, the only truly non-deterministic interpretation. It simply takes the probabilistic nature of quantum mechanics at face value and does not seek to explain it. However, it should be said that the Copenhagen interpretation is generally not (Squires, 1994) considered to be an interpretation at all, but rather a mathematical framework that makes good predictions about a mechanism that is not understood.

### 7.2.4 A possibility from string theory

One of the interesting offerings of string theory is the notion of six or seven curled up dimensions. Clearly, in order for these dimensions to be important in a physical theory, they must have some kind of physical consequences. It is interesting to speculate on whether particles that are identical in terms of the dimensions that we can see, test, and measure, may in fact have differences stemming from the dimensions we cannot see. If this were the case, then things that, from our vantage point, appear non-deterministic, might yet be found to have a cause. Again, however, the real problem arises of experimental proof. If competing physical theories can only be differentiated through measurements that are impossible *in principle*, then it can be argued
that these theories should, more correctly, come under the category of philosophy or metaphysics.

**Answer to Question 7.2: Is physics deterministic (D or [D])?**

From what we have learned in the last sections, we know that this question is still not decided: if we can find a way of differentiating experimentally between the hidden variables theories and the others, then some progress may be made but—for the moment—there is no prospect of this happening. Thus, based on current knowledge either the D or [D] scenarios may be valid.

**Question 7.2b Can the effects of noise be minimized over time (m≠m_i)?**

From our previous discussion, we have established that quantum mechanics is based on probabilities and that their distributions may be learned through observation over time. However, these distributions are not all clustered in a way that is obviously compatible with the basic statistical analysis we discussed in Chapter 5. Take two examples we discussed earlier he time to emission of a photon, or radioactive decay, will have a normal distribution: we can watch the emissions over a long period to get an increasingly good mean value. However, in the case of the photon going through the half-silvered mirror, the answer is binary: it goes left or right.

Here, if we are willing to accept a distribution as a useful answer (it goes left 47% of the time, right 53% of the time), then we can employ the same
statistical techniques to get the best possible information: thus, even if physics is not deterministic, quantum noise can be minimized. If we are not willing to accept this distribution as being sufficient, then the non-deterministic scenario precludes noise minimization.

**Question 7.3: Is the universe bounded or not?**

Interestingly, the issue of the universe being bounded makes little practical difference to our results. Nevertheless, the question has been raised and is worth addressing. There seems to be a broad consensus (see e.g. (Greene, 2000), (Guth, 1997), (Penrose, 1997), (Smolin, 2000)) that the universe started small and is expanding. In fact, this issue (the 'smallness' of the universe at the big bang) is a crucial argument for the necessity of a new theory of physics. It is not yet been decided whether the universe is going to expand forever, or will eventually contract in a 'big crunch' as Penrose suggests. However, this issue is not relevant to our current study: what is relevant that something expanding or contracting is of finite size.

It can be argued that, regardless of whether the universe is bounded or not in fact, it is bounded *practically* for our application. This is because, however large the universe, only a portion of it has been able to effect us because of the limiting factor of the speed of light. As Penrose explains it, you can do a calculation based on the age of the universe and the distance that light can travel in that time. The argument goes that nothing further away could have any effect on our embodied intelligence because the messenger
particles required for an interaction could never reach it (this is the theory of the so-called light cone). If this theory is accepted, then the result is the same as for an expanding universe. The perceived universe (though bounded) gets larger as time progresses, because light (or whatever messenger particle) from further and further away has time to arrive. The same argument can be made for a machine being created or 'born'. It's universe gets bigger as the number of influences that can have reached it during its life increases.

Problems with this latter theory come from the fact that the absolute sanctity of the speed of light may not survive the acceptance of string theory. One prediction (which appears, today, impossible to verify or disprove) is the existence of the tachyon: a massless particle that can move faster than the speed of light (Greene, 2000).

**Answer to Question 7.3 Is the universe bounded?**

Both existing and proposed theories seem to suggest that the universe is bounded, therefore only B scenarios will be considered physically realistic.

**Conclusions from this chapter**

- Quantum mechanics, string theory, quantum gravity, and general relativity do not agree on whether or not space-time is continuous. C and [C] are both plausible.
- Although some interpretations of quantum mechanics suggest that the universe is non-deterministic, there is sufficient dissent and
expectation of change from theories that may supersede it that it cannot be said to be clear cut. $D$ and $[D]$ are both plausible.

• The properties of quantum noise may be learned over time, but—for some types of event—the effects may only be minimized by increasing the accuracy of the known distribution. Thus, $m_1$ may or may not be equal to $m_r$.

• The literature shows little controversy concerning the bounding of the universe. Only $B$ seems to have any physical validity.
Chapter 8:

Discussion of the model's implications and use

Now that we understand the model, its mathematical consequences, and the possibilities given different possible physical realities, we can go back to our original question—do analogue and digital machines differ in their ability to adapt to the environment?—and ask some further questions. For instance, what is the practical value of the physical computation approach to thinking about intelligent embodied agents, and what practical problems does it present? Finally, given the answer to these questions, we will consider where the approach is most usefully applied.

Question 8.1: Do machines differ in their ability to adapt to the environment?

To answer this question we will go through the framework we set up in Chapter 6, modified by our knowledge of Physics from Chapter 7, and summarize what we know.

8.1.1 Real versus virtual interaction

As we have seen, no Turing machine or SLPDC is capable of performing a real interaction in the physical environment: such machines must be considered virtual whichever environmental scenario is true. Thus, it must be the global complementary function, $V^c$, that allows us to ‘feed’ our Turing
machines (or digital computers) with the symbols they need, and to use the resulting symbols to produce an output. We can generalize this to say that any symbolic interaction is, by definition, a virtual interaction, and the global complementary function is crucial to its success.

Analogue or physical machines perform real interaction with the environment as long as that universe is deterministic: this allows an ideally-coupled co-evolution. However, where the interaction is virtual, the global complimentary function is supplied by nature: no additional design work is required to account for it. Instead, the only result is a change in the amount of physical information retained.

### 6.1.2 Physical information retention

As we could predict using information theory, the advantage of analogue continuity disappears as continuity disappears, and analogue machines become Turing-equivalent. This happens in two situations: discretization of the state-space through noise, or through quantum-gravity-related considerations. Thus the perfect coupling of real interaction \((P=1)\) gives way to lower values related to the relative sizes of the two systems and the level of discretization.

As expected, since Turing machines would have to be implemented as real (analogue) machines in our model, the latter always equals or outperforms the former as long as we accept that the noise can be minimized over time. In other words, according to this model, the analogue machine always perform as well as the Turing machine. This may seem unremarkable
until we remember that the Turing machine is a thought experiment. Our model suggests that a noisy or discretized physical computer (it is inappropriate to describe it as analogue when it is discretized) is still as good as an idealized digital computer.

This is in contrast to the SLPDC. By simply putting a limit on the precision (even though we do not explicitly say what it is) the physical information retention of the SLPDC drops to zero. Only where the environment is similarly bounded and discretized does this change. This is interesting in that it corresponds to situations where the conventional engineering approach (as discussed in Chapter 1) would in any case be expected to be more applicable.

**8.1.3 Computational power**

Where true continuity exists (continuous space-time, deterministic physics), so do the super-Turing abilities that Siegelmann describes. Where this continuity is lost, so these disappear.

It is difficult to draw overall conclusions because the physics itself is, currently, inconclusive. Were we to have precise answers to the questions of determinism, continuity, and boundedness, we would be in a position to make clear judgements. For instance, the SLPDC does best in the [C][D] scenarios. If the physics justified choosing these scenarios as the correct one, then the advantage of analogue computation would be marginal at best: it would rely on a practical limit to digital precision. At the other extreme, CD scenarios would suggest that analogue computation has a clear functional superiority. In
the other cases, the analogue-Turing equivalence would offer practical advantage in terms of the lack of an A/D conversion step.

Though for now the physics is inconclusive, if we take all the possible B scenarios into account (as all seem possible currently) then we can look for a dominant trend. One is apparent: that there is nothing to lose with analogue machines, and there is potentially something to gain.

**Question 8.2: What is the practical value of the physical computation approach?**

The inconclusive results in the last section may suggest that nothing has been gained by going through this analysis: at least not unless we can narrow down the options through advances in physics. However, since computers that we can build are either analogue or strictly-limited digital precision computers, we now know that there are specific areas where analogue computation has (potentially) a clear advantage above and beyond those discussed in Chapter 2. By understanding these advantages, we can determine whether it is worthwhile to pursue them because our application demands it: even now when we are not sure whether this advantage will be available or not. It also points to lessons about the nature of machines, their interface with the world, and how we might build both real and virtual machines so that they are as efficient as possible.
8.2.1 Understanding approximation

From Question 6.1.4, we know that there are circumstances where approximating the input during real to virtual conversion (whether this conversion is from the real numbers to the rational, or the rational numbers to a finite set of digital numbers or symbols) will cause an error whose effect cannot be predicted and therefore cannot be guaranteed to be within specific bounds. One of these situations is where the function being performed by the embodied agent is discontinuous and chaotic: in this case, inputs may be arbitrarily close and outputs arbitrarily far. Another is where the error function changes over time in a way that is not known. Since the agent's function is not only determined by it's own learning function but by what it has experienced, the machine's whole lifetime would have to be predicted in advance in order for the error function to be known. Only in this case (and barring the function becoming chaotic at any point) could the precision required be set in advance with confidence that the error would be within a specified bound.

This issue is somewhat reminiscent of the halting problem, in that the only way to know what happens is to run the system. In this case, instead of trying to find whether or not the algorithm halts at a time $t$, the designer is trying to discover whether or not the error goes above a given threshold. Of course, it is not possible to know this unless an analogue machine is running at the same time for comparison.
8.2.2 Taking the system as a whole

Computational neuroscience (the study of biological neural structures using tools more commonly associated with computer science and engineering) often uses information theoretic approaches to understand the brain. In a paper like (Abshire & Andreou, 2001), for instance, Shannon's communication theory is used to improve our understanding of the vision system of a blowfly. It does this by considering the chain of events triggered by a particular circumstance (such as a number of photons hitting the retina in a given time period) and then looking mathematically at the transformations that the information undergoes at each step to see how it is lost.

This method is very helpful in that it allows us to think about individual neural systems at the microscopic level. As we look at them and see structures that encode, and thus retain, information well, we can learn a great deal about why evolution has designed them this way. In areas of the brain that are chiefly concerned with light, for instance, it is not at all surprising to find structures that, from an information-theoretic standpoint, are particularly good at handling light signals.

However, the brain is not a well-insulated, serial, electronic system, and the problems that it solves are not well-insulated either. For instance, mutual information (see e.g. pp. 40-41 of (Deco & Schürmann, 2001)) often exists between several different sensory modalities, and the information coming into the system through a single channel may have more than one cause. Related to the latter phenomenon, independent component analysis (ICA (see, e.g.
(Comon, 1994) is a well-known means of separating audio input comprised of many different signals into its source components. It generally requires a number of different sensors in order to work (say a microphone per source in the audio case). Essentially, ICA is a more thorough version of principle component analysis which works through correlation of the incoming signals. In the case of multi-sensory information, there are artificial systems (based on mutual information, see, e.g. (Becker, 1996)) that can attribute a single cause for information coming through different channels (de Sa & Ballard, 1998): in this case, audio and visual.

Given these two abilities, both attributed to neural systems, a third naturally arises that has yet to be fully explored: the ability to separate information from several different sensor modalities travelling down single channels. In other words, if the neural response of the eye is somewhat modulated by pressure waves from sound, and the response of the skin and ears are modulated by temperature, etc., then each of these minor sensor modalities (those that are not the main target for a given sensor) would normally be seen as noise corrupting the main type of information carried down a particular sensor channel. However, there is no reason in principle why these so-called noise components cannot be extracted out to reinforce another signal.

Thus, information theory is helping us to understand the way various structures work (and so helping us to think about how to design artificial versions). On the other hand, it is limiting us to a relatively narrow view of the
functionality of any particular system. We can think, for instance, of the visual system as purely dealing with incoming light as a kind of first-order approximation of its total contribution. Combining this with factors related to the other senses might be considered a second-order approximation. But the danger in using such an analysis to define limitations on the global properties of the brain are clear: it may be processing more kinds of information than we realise. As a result, signal changes that we label 'noise corruption' may, in fact, be signs that information is being added, not taken away.

This is an argument that suggests that the whole will always be more than the sum of its parts. Not because the analysis of any of the parts are incorrect, but because they cannot all be combined into a single model: in simple terms, the number of simultaneous equations would make the problem intractable. Such limitations are on the model, however, and not necessarily on the physical brain itself. This would be true whether the brain were real or artificial.

A very different argument extends the idea of the whole from the intelligent system alone to the system and it's environment: in the sense that we shape our environment to aid our survival and ease our ability to interact with our surroundings (e.g. (Clark, 1997) or (Russell, 1993) for a specific example and application to robotics). Though it does not help us with the details, the physical computational approach is clearly sympathetic to the importance of situatedness in intelligence.
8.2.3 Providing footholds for evolution

The complex structures that exist in biology can, as discussed in the last section, be considered in information-theoretic terms. But how did these structures evolve? Mammals with sophisticated light or sound-sensing organs evolved from single-celled organisms. We know that this happened through evolution. But how does a sense organ evolve when there wasn't one there before? By considering the entire physical object as a sensor, just by virtue of the fact that it obeys the laws of physics, this model allows nominal sensor effects caused by physical structures to be exploited.

This can be thought of in two ways. First, you can consider the effect in a single animal. Physical computation, for instance, explains the ability of deaf people to hear through their bodies as the percussionist Evelyn Glennie has learned to do (Glennie, 1996):

Evelyn spent a lot of time when she was young (with the help of Ron Forbes her percussion teacher at school) refining her ability to detect vibrations. She would stand with her hands against the classroom wall while Ron played notes on the timpani (timpani produce a lot of vibrations). Eventually Evelyn managed to distinguish the rough pitch of notes by associating where on her body she felt the sound with the sense of perfect pitch she had before losing her hearing. The low sounds she feels mainly in her legs and feet and high sounds might be particular places on her face, neck and chest.
Thus, physical features that are not designed as sensors can nevertheless be exploited as such. This is only possible, because—in the physical computation model—no explicit measurement path is required for these non-sensors. If it were, in an engineered system there would almost certainly be no such path. The kinds of vibrations involved in hearing, even at low frequency, would be unlikely to be considered of primary importance in an information-theoretic analysis of a leg: they would be considered too marginal (in terms of purpose) and too small (in terms of signal) to be worth collecting. Nevertheless, the brain can make use of the information that the legs provide as hearing organs: and in Glennie's case it actually does.

Such small effects can also be exploited by changing the entire morphology of a species. For instance, there is much debate about the reason for the shape of the hammerhead shark (Kajiura, 2001). Unlike an ordinary shark, this animal has an oblong-shaped head much wider than its body wide ends at the front and neck with the eyes on the front two corners of the oblong. There are various reasons why biologists believe that this shape evolved. First, there is an obvious sensory advantage in that binocular vision (depth perception) improves with the separation of the eyes. Another explanation is that this head shape has hydrodynamic advantages including increased stability while turning. Finally, sharks have electrosensory pores that assist them in catching prey. The wider head means there can be a greater number of these pores for the same density.

With the physical computation model, evolution could select for any of
these properties (i.e. they would supply some kind of advantage that would allow the animal to survive and procreate better than its neighbours) because any physical changes caused by mutation, no matter how small, or how seemingly unrelated to either sensing or locomotion, could be exploited. For instance, a slight widening of the head produced by evolutionary accident, could have improved the survival of the hammerhead shark precisely because all three abilities (electro-sensing, depth perception, dynamical stability) were enhanced. Similarly, we can imagine that a hair-like structure—one that happens to vibrate at a frequency of interest and happens to be located near some kind of nerve ending—could eventually evolve into a hearing organ.

Another example of this might be the vision system of the sunfish (Cameron & Pugh, 1991). This creature lives in areas where the water is relatively cloudy, causing much diffusion. However, the morphology of the eyes allows polarization-difference imaging: where images are acquired of each polarization (due to the shape and orientation of the photoreceptor anatomy) and then subtracted from each other. This removes most of the diffuse light, leaving a useful image for the fish to interpret. Again, in terms of evolution, the physical computation here is crucial.

From an engineering perspective, development of the physical computation approach could eventually allow the use of genetic algorithms to select for success in a much more powerful way. Though evolutionary algorithms initially focussed software agents in artificial environments (Levy, 1992), the idea of engineering hardware this way has been around for some
time, initially working with simulated physics (Harvey et al. 1994; Husbands et al. 1997). Later, from the same group, Thompson (Thompson, 1999) showed that small hardware effects can be exploited by GAs to solve problems in ways that would not have been possible using conventional circuit design techniques or software-based design packages. Also, Sims (Sims, 1994) demonstrated that it is possible to engineer brain and body together, with physical and informational attributes evolving together in a competitive genetic selection process (he used a virtual physical model of the environment to do this). By combining all three approaches, we may be able to evolve embodied agents that can multitask, using their physical attributes to best effect in terms of sensing, actuating, and processing.

8.2.4 Processing the unmeasurable/computing the incomputable

If the most optimistic scenarios for analogue computation are applicable, the CD scenarios, physical computation has advantages over Turing machines. Literally, this means that analogue could compute the incomputable, where computable is defined as computable on any discrete-state machine. Also, real-interaction-based physical computation does not require the making of measurements in the way that virtual computation does. Consequently, physical parameters that could never be known, even in principle, can nevertheless contribute to the physical processing taking place. In an application where sensitivity to the environment is an advantage, this is an important property.
It is important to stress at this point that this advantage cannot be easily harnessed (if at all) for conventional computing applications like word-processing or database management. The advantage is derived by two things that make it incompatible with conventional sensor-processor-actuator systems. First, the ambiguity of the sensors (the whole device is a sensor) means that all physical influences can be taken account: however, their sources cannot all be separated. For instance: we are subject to gravity, and any small changes in its force on us will affect the way we behave. However, that does not mean we know about all the different masses that have combined to create this particular force. Such ambiguity—while acceptable for applications where the macroscopic physical behaviour of a system is what's important—cannot supply the information required for conventional informational tasks.

8.2.5 The analogue shell and nested virtual machines

Chris Toumazou (Toumazou, 2003) has talked about the analogue shell of a device: the outer sensors and actuators that allowed the inner digital electronics to interact with the outside world. The physical computational model fits well with this metaphor. Though Toumazou does not work on artificial intelligence, the systems that he designs—which include mobile phones and biomimetic sensors—fit well into a hierarchy that is suggested by the model. There are three obvious examples we can discuss: the digital virtual machine, the analogue virtual machine, and the analogue real
machine. The latter should be well-understood by now as it has been the subject of much of the discussion so far. However, the other two cases are worth further consideration.

For virtual interaction, the first extra cost is that of shielding: in order to be selective about the information acquired (information implies selection) the object must be resistant to the other physical influences around it. Second, the object must interpret the information coming in through its sensors (and information it is, in this context) and assign it some kind of meaning (even if this is purely in the context of which bit comes before which) before it can act. As discussed in Chapter 2, this tends to mean dramatically reducing the types of information that can be considered and the amount of information that can travel through those channels. For instance, we might consider visual information (which is really a kind of map of our local electro-magnetic field) important, then reject certain wavelengths for consideration because their source, for instance, is ambiguous. Without the relevant knowledge about the choices made, the signal—or information—becomes meaningless.

We can understand these shielding and interpretive functions as being part of the complimentary function, $V^C$. Where the virtual intelligence function is the digital (or analogue) processor, the global complementary function is the machine’s analogue sensors and actuators and enablers: mechanics, optics, hydraulics, heat-sinks, and so forth. It is important to remember here that just because a machine is analogue, doesn't mean that its interaction is
real (even if real interaction can take place). If we are running an information-theory-based system and expect information-theory-based results, then the interaction with the environment is virtual.

In fact, the electronics industry is entirely structured on the basis of this duality. On the one hand, designers treat resistors, transistors, capacitors, etc., in terms of mathematical functions that they will perform on an expected (in terms of dynamic range, frequency, etc.) signal from a known source. Software engineers do not even consider the implementation: they simply assume that the high-level functions they need will be permitted on the machine they use. At the lowest level, a completely different set of designers look at the physics of the systems that will make both of these higher levels work efficiently. You can think of this as the analogue shell (complimentary function) with a first virtual layer of the designed electronics and the second virtual implementation of the software. These levels have clear demarcations.

What is potentially interesting about an all-analogue implementation is that there may be many levels of virtual computation rather than the two we've just described for digital. In an information-theoretic sense, we can think of many layers of analogue circuits, each one throwing away more information from the outside world in favour of those signals that are deemed important (by virtue of, for example, neural network learning). It could be argued that the visual system, as described, for example, by Hubel and Wiesel (Hubel & Wiesel, 1962)) does something like this. Rather than encode all information, retinal processing elements seem to break down images into lines of different
orientations and so forth. This concept of filtering is also used in pattern matching through Fourier-transform-based image correlation. Instead of comparing the images, elements of the image are filtered out and considered as a collection of elements. The more closely the collection of one image matches another, the more closely the images themselves are expected to match.

We can consider adaptive multi-level filtering as a kind of nested virtual machine: where an agent takes as much information into the system as is feasible given the sensors and then deciding what to throw away based on experience. This is a much more flexible system than one that requires immediate (and irrevocable) analogue-to-digital conversion. And yet it is compatible with conventional artificial intelligence in that it allows for a pseudo-symbolic core to emerge. Thus, conventional algorithms that people have devised to explain intelligent behaviour in people may be considered accurate: the question is how to implement them within a nested virtual system implemented on an analogue machine and without the usual interface of software.

8.2.6 Intelligence is more than thinking

The concept of the complimentary function may be helpful also when considering the mind-body problem (from a materialist standpoint). Our bodies not only provide our conscious mind (the intelligence function in this case) with sensors and actuators, but also the un/subconscious mind which is
not normally considered in AI. Other bodily systems—the limbic system, the spinal chord, even basic organs like the heart and lungs—all contribute to our behaviour to a greater or lesser extent. Using the virtual interaction model, we can see that what many people consider to be artificial intelligence, particularly purely software-based approaches, concerns $\mathcal{V}$ and not $\mathcal{V}^c$.

This balance may also be said to be the concern of the neuromorphic engineers, discussed in Chapter 2: their work often goes much further, blurring the interaction boundaries. Projects like Harrison and Koch's all-analogue fly vision system and robot controller (Harrison & Koch, 2000), Hasslacher and Tilden’s analogue walking/light sensitive robots based on the nervous systems of small animals (Hasslacher & Tilden, 1995; Hasslacher & Tilden, 1997), or Lewis’s bipedal robots based on central pattern generators (Lewis & Simô, 1999; Lewis et al. 2001), all have in common that there cannot be said to be a clear line between sensor, processor, and actuator. Indeed, there cannot be said to be a clear line between software and hardware.

Pushing this to an extreme, one could argue that full understanding of real interaction, virtual interaction, and the global complementary function might go some way to addressing Brooks’ recent question about the relationship between matter and life (Brooks, 2001). In essence, we may just need to go further to try to understand a whole brain and body (both $\mathcal{V}$ and $\mathcal{V}^c$) in order to understand how a creature works, rather than considering mind or brain processes alone. Considering this issue in the 1940s,
Schrödinger suggested that the secret to the difference between what is and isn't alive lies in the nature of its reproduction and the coding of its construction in genetic code (Schrödinger, 1944). More recently, Pattee has suggested the same thing, and pointed to the importance of self-knowledge (in the sense of what is, and isn't, the self: see e.g. (Pattee, 2001), and the importance of physical symbols (sets of physical states that essentially 'mean' the same thing). Since the integrity of both the genetic code and the self (whether at the level of single-celled organisms or animals with a large and complicated structure) is based on physics and chemistry, a more material, less abstract, understanding of these issues may be helpful.

**Question 8.3: What are the difficulties of the approach?**

Although the physical computational approach may provide us with new conceptual avenues down which to pursue embodied artificial intelligence, its adoption has three major problems associated with it.

**8.3.1 Design and fabrication**

Complicated physical systems are notoriously difficult to design and can be expensive to build. According to industry experts, too few analogue engineers are being trained (Briggs, 1999). There may be several reasons for this. First, the mathematics involved in analogue electronic design is much more difficult than digital. Second, even if a designer (particularly at the academic level) has the right analogue skills, he or she may have to wait a long time (weeks) for
designs to come back from a VLSI foundry. Third, a relatively small design mistake or miscommunication could render the manufactured chip useless. Finally, the process is as expensive as it is time-consuming. A similar argument could be made for the work of mechanical engineers and optical engineers. By contrast, programming (software engineering) is relatively easy, inexpensive, and can be done by a single researcher without assistance or delay.

Though research progresses towards simplifying the design process of optical, mechanical and analogue systems (see, e.g., (Haigh, 2004)), the level of difficulty is currently still vast in comparison to the ease of sitting down to programme a computer.

8.3.2 The unpredictable black box

There is another problem that comes through working with any machine that has an interface to the real world and which is not well understood: unpredictable behaviour. In a desk- or laptop computer this is not a problem because, in most circumstances, it poses no safety issues. In an embodied intelligent agent—whether one designed to have the same physical abilities as a human, or an autonomous aircraft, or a power plant—this is not the case. The larger and more potentially destructive a machine is, the more predictable we generally want it to be.

While any embodied AI is likely to be complicated and therefore difficult to predict, a neural-network-based analogue AI is likely to be much more so.
This is because the behaviour of the machine is determined by its history alone, and not by known algorithms designed by software engineers. In a sense, this could make the machines more human, as people are also considered to be black boxes, conditioned by their experiences. On the other hand, it would not seem to be worth risking such an unpredictable approach for straightforward engineering applications or those in which it is easier to simply have a human operator in control.

8.3.3 Implementation, not explanation

The physical computational approach is intended to allow researchers to think about the most efficient possible solution to a problem in a given physical environment. However, like other approaches such as genetic algorithms (natural selection), it doesn't seek to explain why a particular solution is better than another. In fact, the parallel with natural selection is very close. Both it and physical computation explain the mechanism by which intelligent evolution takes place in an environment, not what form it should take. Thus, conventional models such as those based on information theory have two advantages. First, they can explain why a solution is a good one. Second, if they are sufficiently small problems that they can be adequately computed on digital hardware and with a reasonable analogue to digital conversion, such models are much more cheaply and easily implemented than the physical computational approach would be.

This ease of implementation, along with the ability to devise and then test
hypotheses about the way intelligence works, makes the conventional approach very attractive. It is only where efficiency and sensitivity (either physical, informational or both) are crucial that the physical computational approach becomes worth the extra effort.

**Question 8.4: When should the physical analysis be used?**

There is an appropriate physical analogy in wave-particle duality. A pulse of light can be thought of as both a wave (actually a large set of waves of different frequencies superimposed on each other) and a particle (a localised energy distribution) at the same time. The reality of each becomes apparent in different contexts: for instance, individual frequency components of a single wavelength can be used to record individual holograms of different colours (Wild et al. 1985); but pulses of a particular wavelength can also supply energy to trigger a particular detector or optical switch.

In a similar way, physical objects such as computers can be analysed using either method. The physical method can be compared with looking at the position and behaviour of the light-pulse in space, whereas the communications theory approach allows for the consideration of the individual waves, infinite in extent, that combine to make the whole. Each wave could be thought of as a different kind of informational path (specified, for instance, by the physical quantity or quantities involved and the perceived input/output mechanisms). In a particular physical object, the set of all such paths might include a programme running on the implementation of the Turing machine.
For applications where picking out these individual strands is useful, the trick would seem to be knowing how to isolate the informational path of interest: by knowing what constituted an input or output for this particular path.

In this way, a notebook computer may be understood with sufficient understanding of the inputs (keyboard/memory etc.) and output (screen, printer) and the significance of the symbolic coding. It may also be understood as a simply a mechanism for turning electricity into heat and light, it's ability to turn kinetic energy into sound (keyboard), and so forth. By adding up this arbitrarily large and potentially infinite set of components, communications theory could describe the physical object: but it is unlikely to be the most efficient way to describe it. Likewise the physical description could be analysed, but with great difficulty, to understand the Turing-machine operating within.

Embodied human-like artificial intelligence is a particularly good example of an application where information theory would not seem to be sufficient. Just looking at an individual interaction channel (like, for instance, the machine's position on a 2D map and proximity to other objects) is insufficient. The machine's embodied nature, encompassing its brain, means it has a myriad of ways of interacting with the world. Looking at a few of these will only tell a small part of the story. They may provide a helpful explanation, but will not be able to explain global behaviour.

Interestingly, the reason that Hava Siegelmann, who developed the super-Turing model of recurrent analogue neural networks, started her work, was due to this kind of argument. In discussions she said that it was not that
she wanted to prove that such networks were super-Turing. Rather, she wanted to warn those who might want to compute with them that, if they used the conventional tools, they would be computing something different than they would have expected (personal communication).

So, the issue is raised, are the design rules for building these physical systems different from building informational ones? The answer is probably no, in that information-based tools are all we have. However, knowing their limitations may lead us to find new ways of testing for a the class of emergent properties related to 'hidden' interactions. It can be argued that by moving from the basic science of physics to disciplines working at different levels of complexity (such as chemistry and biology) we are precisely recognising the shortcomings of analysis on the wrong scale. However, further consideration of this question is beyond the scope of this thesis.

Conclusions from this chapter

- Depending on the view one takes of physics, analogue machines, Turing machines, and SLPDCs may have different levels of information loss, experience different types of interaction, and different kinds of computational power. Analogue is generally as good or better than Turing, which is generally better than the SLPDC.
- The physical computational approach has the advantages of providing us with: a better understanding of the concept of approximation; a global understanding of a system's abilities; a more solid foothold for
evolutionary design involving physical shape and function; potential super-Turing computation; and a better understanding of the concept of the virtual machine.

- Design difficulties, the expense of implementation, the unpredictable behaviour of truly adaptive analogue systems, and the fact that physical computation does not necessarily provide insight into mechanism, all present problems for the approach.

- The physical analysis is most appropriately used for large, multi-tasking, poorly-understood problems—such as embodied AI—where adaptation to the environment is important.
Chapter 9:

Conclusions and future work

In this thesis we have addressed the issue of physical computation, presented a model of it, and shown its consequences. The conclusions from this work are shown below.

Conclusion 1: The Turing-machine and other symbolic models cannot adequately address embodiment (see Objective 1, p. 2)

As we discussed in Chapter 2, digitally-implemented neural networks do not scale well compared to highly-interconnected networks for the discrimination of environmental features. Nor are they able to perform well in applications with high computational complexity, where analogue networks are both smaller and more efficient than their digital counterparts. Further, systems that use digital processing require A/D conversion, which requires that some information be thrown away. Finally, we have seen that, where sensing, processing, and actuation are done together, the speed, energy efficiency, and degree of sensor fusion may be increased.

In Chapter 3 we saw that, in any case, Turing machines have limitations that may preclude the implementation of embodied artificial intelligence: not least the fact that they cannot interact with the environment directly. They cannot, for instance, simulate physical systems to arbitrary accuracy if physics
involves the performance of chaotic functions. Recurrent analogue neural networks, on the other hand, can theoretically perform super-Turing functions in certain circumstances, though some of these abilities seem to erode with noise.

9.1.2 The physical computation model can accommodate both physical and Turing computation (see Objective 2, p. 2)

In Chapter 4, we set up a model that addresses the ability of physical objects to sense, actuate, and learn. By considering the different types of interaction (real and virtual) between a physical system and its environment, we can more fully understand the relationship between physical reality and a virtual machine: the latter is a construct that arises by considering a subset of inputs and outputs of the former.

In Chapter 5, we then showed how known types of virtual machine—analogue, Turing, and digital—can be defined to fit into our model, and how the effect of noise may be different than is classically considered.

9.1.3 Physics is the only necessary constraint on computation (see Objective 3, p. 2)

To understand how different types of machine can perform, we saw in Chapter 6 that we must have information about the nature of the physics under which they operate. In particular, we must know whether space-time and other physical dimensions are continuous, whether physics is
deterministic, and whether or not the universe is bounded.

When considering these issues, we found that analogue, Turing, and strictly-limited-precision digital computers may have abilities to retain information even whilst operating in virtual mode. The actual capacity is determined not only by the physical scenario, but by the minimum sizes of any discretizing element (whether noise or intrinsic quantization), and the relative sizes of the object and environment. We found that the physical information retention for analogue machines is always as good as, or better than, that for Turing machines, which is always as good or better than that for SLPDCs.

Finally, virtual interactions (to which symbolic machines are restricted) may approximate real physical interaction to a given precision only where the embodied agent's function is both continuous and known. If it is chaotic or unknown as it varies with time, then the maximum-allowed input error required for the approximation cannot be calculated. For noisy analogue machines, the function need not be known.

9.1.4 Today's physics does not provide conclusive answers (see Objective 4, p. 3)

As was discussed in Chapter 7, space-time cannot be conclusively said to be continuous (or not): quantum mechanics, string theory, quantum gravity, and general relativity disagree. Likewise, different interpretations of quantum mechanics disagree on whether or not physics is non-deterministic, and the whole field should eventually be superseded in any case. Only the issue of the
boundedness of the universe seems to be generally agreed upon: it is bounded.

9.1.5 Physical computation is useful, primarily, for large and ill-posed problems (see Objective 5, p. 3)

As we discussed in Chapter 8, analogue machines, Turing machines, and SLPDCs may have different abilities to retain physical information and interact differently. Independent of the inconclusive physics, we can say that analogue machines are as good or better than Turing machines, which are better than SLPDCs. Further, the physical computational approach gives us a better understanding of the concept of approximation, a more global understanding of a system's abilities, a more powerful use for evolutionary design, and potential super-Turing computation. Further, it solidifies the concept of the virtual machine, unifying different-level hardware and software approaches.

Practically, however, taking this approach increases the design/implementation difficulty and cost for some applications, as well as introducing the likelihood of unpredictable behaviour. Further, though a device designed using such an approach may work well, it may nevertheless be impossible to understand (a 'black box'). Thus, the physical computation model is most appropriately used for large, multi-tasking, poorly-understood problems—such as embodied AI—where adaptation to the environment is important.
9.2 Future work

There are several avenues that have yet to be fully explored. In particular, we would like to be able to produce a diagnostic test for an engineer to determine whether or not the physical computation approach is necessary for his or her application, and a set of design rules to use if it is. These would specify how analogue to digital conversion layers are used in a given system: i.e. how to correctly balance the analogue and digital processes in order to maximize efficiency for a given task. In some cases it might be expected that no conversion is the best option, thus suggesting an analogue-only system.

Further, we believe that a review of different types of hardware (different types of electronic, optical, mechanical, thermal, etc. systems) would be useful. Such a review would consider where different types are devices are potentially exploitable in analogue systems, and how they would be most usefully structured.
References

Abshire, Pamela and Andreas Andreou (2001), A communication channel model for information transmission in the blowfly photoreceptor, BioSystems 62, pp. 113–133.


Anderson, Herbert L. (1989), A Physicists Desk Reference, American
Institute of Physics.


Briggs, Suzy (1999), *ON Semiconductor Provides $1.5 Million to Support Analog Electronics Program*, ECE News, 1.

Bringsjord, Selmer and Dave Ferucci (1999), *BRUTUS and the Narrational Case Against Church's Thesis*, Thematic talk presented at AAAI Fall Symposium on Narrative Intelligence, 5-7 November.


Brooks, R. A. and L. A. Stein (1993), Building Brains for Bodies, AI Memo No. 1439, Massachusetts Institute of Technology Artificial Intelligence Laboratory.


Duhamel, Jean-Rene, Carol L. Colby, and Michael E. Goldberg (1998), *Ventral Intraparietal Area of the Macaque: Congruent Visual and


Fairchild Semiconductor (2003), *Fairchild's Optically Isolated Error Amplifier Features Tolerance as Low as 0.5% in a Space Saving SOIC Package*.


Farhat, Nabil H. and E. del Moral Hernandez (1996), *Recurrent networks with recursive processing elements: paradigm for dynamical computing,*


http://www.digitalphilosophy.org/finite_nature.htm


Gandy, Robin (1980), *Church's Thesis and Principles for Mechanisms*, in *The
Kleene Symposium, North Holland Publishing Company.


http://www.evelyn.co.uk/hearing.htm

Gödel, Kurt (1931), On formally undecidable propositions of principia mathematica and related systems, Dover.


2291–2304.


Hecht, Eugene (1987), Optics, Addison Wesley.


Jokerst, N. M., Martin A. Brooke, Joy Laskar, D. Scott Wills, April S. Brown, Olivier Vendier, Steven W. Bond, Jeffrey B. Cross, Michael Vrazel, Mikkel Thomas, Myunghee Lee, Sungyong Jung, YoongJoon Joo, and Jae J. Chang (2000), *Smart Photonics: Optoelectronics*


Kugel, Peter (2003), *If intelligence is uncomputable, then*, Special Session on Beyond the Boundaries of Computability presented at *Wester Section Meeting of the American Mathematical Society,* May 3-4.


Landman, Bernard S. and Roy L. Russo (1971), *On a pin versus block


Mok, Fai, Demetri Psaltis, and Geoffry Burr (1992), Spatially- and Angle-
References


Penrose, Roger (1989), The Emperor's New Mind, Oxford University Press.

Penrose, Roger (1994), Shadows of the mind: a search for the missing science of consciousness, Oxford University Press.

Penrose, Roger (1997), The Large, the Small, and the Human Mind, Cambridge University Press.


Rodgers, Peter (2004), Hawking loses black hole bet.

http://physicsweb.org/article/news/8/7/11/1


Schrödinger, Erwin (1944), *What is Life*, Cambridge University Press.


Sims, K. (1994), Evolving 3D morphology and behavior by competition, in
Artificial Life IV, MIT Press.


Tegmark, Max (2000), Why the brain is probably not a quantum computer, Information Sciences 128, pp. 155–179.


Appendix 1:

Glossary of symbols

The following are arranged in alphabetical order: first by Latin alphabet, then by Greek, and finally by Hebrew. This glossary is intended as an aide-memoire only: for full definitions, please refer back to the text.

Latin:

A The set of analogue functions.

A An analogue function, $A \in \mathcal{A}$.

B Condition where the universe is bounded.

[B] Condition where the universe is not bounded.

C The set of computable numbers. Includes the rational numbers (from $\mathbb{Q}$) and those irrational numbers (from $\mathbb{Q}'$) that can be represented by finite means.

C Condition where space-time is continuous.

[C] Condition where space-time is not continuous.

d Minimum space-time chunk size in a discrete universe.

D The set of analogue values that can be distinguished (known with given confidence) in the presence of noise.

D Condition where physics is deterministic.
Appendix 1: Glossary of symbols

[D] Condition where physics is not deterministic.

$E$ The environment function with the change of state incorporated as a dependence on $t$.

$E$ The environment function, dependent on $\psi_t$ and $y_t$.

$F_{in}$ The function that maps $x$ to $x'$, or to $x'$ and $x''$.

$F_{out}$ The function that maps $y'$ to $y$, or $y'$ and $y''$ to $y$.

$G_t$ The intelligence function with the change of state incorporated as a dependence on $t$.

$G$ The intelligence function, dependent on $S_t$ and $x_t$.

$G$ The set of all possible intelligence functions, $G$.

$I$ The set of integers.

$L_p$ The laws of physics.

$L_V$ Virtual learning function. Determines how state $S'_t$ changes depending on $x'_t$.

$L_V$ Virtual learning function. Determines how $V_t$ changes with time.

$l_\Theta$ The size (length) of a bounded, one-dimensional environment.

$L_\Theta$ Environment learning function. Determines how physical state $S_t$ changes depending on $x_t$.

$L_\Theta$ Environment learning function. Determines how $E_t$ changes with time.
\( l_\Sigma \)  
The size (length) of a bounded, one-dimensional, intelligent system.

\( L_\Sigma \)  
System learning function. Determines how physical state \( S_i \) changes depending on \( x_i \).

\( L_{\Sigma} \)  
System learning function. Determines how \( G_i \) changes with time.

\( m \)  
The gradient of a straight line.

\( m \)  
The number of outputs coming from the environment.

\( n \)  
The number of inputs coming into the system.

\( \mathcal{N} \)  
The natural numbers.

\( \mathcal{N}_{\text{max}} \)  
The cardinality of a set with an unknown (but finite) number of members, as determined by a designer.

\( \mathcal{N}_{\text{MAX}} \)  
The cardinality of a set with an unknown (but finite) number of members, as determined by physics.

\( N_p \)  
The number of inputs possible for an entire population.

\( N_S \)  
The number of inputs in a sample taken over a given measurement time.

\( P \)  
The information retention for a given \( F_{in} \).

\( P_x \)  
The probability that a virtual input \( x' \) came from a given \( x \).

\( Q \)  
The set of rational numbers.

\( Q' \)  
The set of irrational numbers.
\( \mathcal{R} \)  The set of real numbers or continuum.

\( S \)  The set of functions, \( S \), that can be performed on a strictly-limited-precision digital computer.

\( S \)  Strictly-limited-precision digital functions, \( S \in S \). Those that can be performed on a strictly-limited-precision digital computer.

\( S_\tau \)  The physical state of the embodied system at time \( \tau \).

\( S_\tau' \)  The state of the virtual machine at time \( \tau \).

\( t \)  Time, which may be continuous or discrete.

\( \mathcal{T} \)  The set of all possible Turing functions, \( \mathcal{T} \).

\( T \)  Turing functions. Those functions can be performed on a Turing machine, \( T \in \mathcal{T} \).

\( V \)  The virtual function, dependent on the state, \( S_\tau' \), and \( \tau \).

\( V \)  The virtual function, with the change of state incorporated as a dependence on \( \tau \).

\( V^c \)  The global complementary function, \( F_{in} V^c F_{out} \).

\( V^c \)  The complementary function.

\( w \)  The width of a given confidence interval.

\( x \)  Output from the environment to the system, \( x \in \mathcal{X} \).

\( x' \)  The input to the virtual machine, \( x' \in \mathcal{X} \).

\( x'' \)  The input to the complementary function.
Output from the environment to the system at time \( t \).

\( X \)  
The set of all possible inputs, \( x \), to the system.

\( X' \)  
The set of all possible inputs, \( x' \), to the virtual machine.

\( y \)  
Output from the system to the environment, \( y \in Y \).

\( y' \)  
The output from the virtual machine, \( y' \in Y' \).

\( y_t \)  
Output from the system to the environment at time \( t \).

\( Y \)  
The set of all possible outputs, \( y \), from the system.

\( Y' \)  
The set of all possible outputs, \( y' \), from the virtual machine.

\( \alpha \)  
The function that maps the system physical state \( S_t \) to the time-dependent intelligence function \( G_t \).

\( \beta \)  
The function that maps the environment physical state \( \psi_t \) to the time-dependent environment function \( E_t \).

\( \gamma \)  
The function that maps the virtual state \( S'_t \) to the time-dependent virtual function \( V_t \).

\( \Delta t \)  
The time period over which a measurement occurs.

\( \Delta \tau_r \)  
The time period between light pulses as perceived by the receiver.

\( \Delta \tau_r^{\text{pop}} \)  
The mean period between received light pulses for the entire population.
$\Delta r^x_t$ The mean period between received light pulses over a given sampling time.

$\Delta \tau_s$ The time period between light pulses as dispatched by the sender.

$\partial t$ The time-dependent error function. Describes how the error varies with time.

$\partial x$ The difference between input $x$ and the $x$ it maps to.

$\epsilon$ The difference in the output $y$ and $y'$ because of a given $\partial x$.

$\Theta$ The environment of the system.

$\Phi_t$ Analogous to the more commonly written $\partial t$, or $\Delta t$. An incremental amount that can be used to indicate both continuous and discrete time.

$\sigma$ The standard deviation for a population.

$\sigma_m$ The standard error of the mean.

$\Sigma$ An embodied system.

$\psi_t$ The physical state of the environment at time $t$.

$\chi$ The confidence level for a measurement.

**Hebrew:**

$\mathcal{N}_0$ The cardinality of a countably-infinite (denumerable) set such as the integers or rationals.
The cardinality of an uncountably-infinite (non-denumerable) set, such as the continuum or real numbers.
Appendix 2:

Sunny Bains

*Physical computation and the design of anticipatory systems*

Int'l Conf. on Systems, Man and Cybernetics

The Hague, 10-13 October, 2004
Physical computation and the design of anticipatory systems

Sunny Bains
Intelligent Systems and Networks Group
Department of Electrical and Electronic Engineering
Imperial College London, UK
and Department of Design and Innovation
The Open University, Milton Keynes, UK
sunny@sunnybains.com

Abstract - For a system to learn from experience and predict the future, it must have access to the appropriate information. In conventional computational systems engineering, signals of specific types are sought at specific resolutions. Though this is appropriate for many well-understood problems, it may be inappropriate for other, less-well-defined, applications. These include multi-tasking anticipatory systems that interact with the real physical world. Here, the engineer may not have enough advance knowledge to determine what information will be relevant and what will not, particularly in the context of systems that perform complex tasks. Here we discuss how our physical model of computation can allow information to be used more effectively by either postponing, or forgoing, analogue-to-digital conversion. Further — based on evidence from neuromorphic engineering and theoretical computer science — we suggest that systems designed using the approach will be more efficient, both energetically and computationally, than their conventional counterparts.

Keywords: Neuromorphic engineering, analogue electronics, embodied artificial intelligence.

1 Introduction

We have developed a model of physical computation that is intended to do three things. First, we wanted to find a way of thinking about building embodied, multi-sensory, intelligent systems — such as we are — that could encompass both theoretical and physical approaches. In particular, we wanted to build a bridge between analogue neuromorphic engineering, as pioneered by Carver Mead [23] and theoretical work by Hava Siegelmann [27] suggesting that recurrent analogue networks (like those in our brain) have super-Turing properties: that is, they could do everything a Turing machine could do and more. Second, we wanted a model that our model had an interface with physics, so that as our knowledge improves, we can simply choose the instantiations of our model that best fits our newfound understanding. Finally, we wanted a model that could be considered to unify the symbolic with the physical: i.e. the Turing-machine with its implementation.

In this paper, we intend to describe the new model only briefly: full details and motivation can be found in [2]. Instead, we intend to concentrate on the potential utility of the model, how it can change the way we look at anticipatory systems, and what its potential drawbacks are in real application. In particular, we will consider how thinking about computation as physical, rather than informational, may give us a broader picture of what is going on: one that is more consistent with the existence of emergent properties in evolutionary settings, whether external (evolution of species) or internal (evolution of a single brain). On the negative side, we will consider the difficulty — for engineers — of both designing and predicting the behavior of systems using this approach.

2 What is physical computation?

Physical computation is a simple model that can be used to analyze any kind of physical object: from a human to a glass of water to a laptop computer. The assumptions upon which it is based are simple: that any change in the environment (everything except our physical object) will cause some kind of change in the object itself, and vice versa. As both object and environment are restricted to obeying the laws of physics, any intelligent or anticipatory behavior on the part of the object may only be attributed to its physical structure, which changes in a seemingly intelligent way based on the energy, matter, fields, etc., that impinge on it. When the object is coupled to the environment in this way, we call it real physical interaction.

However, in engineering — and even in science sometimes — we do not consider all inputs and outputs when interpreting a behavior. So, for instance, pushing the key for the letter A in a hard or soft way is considered the same input although, physically, it is completely different. Likewise, the size, brightness, and font of the letter on the display do not alter our interpretation of it. This we call virtual interaction because we are interacting with a virtual world, not a real one. Virtual interaction requires that the physical inputs be filtered, or otherwise mediated. Because this mediation takes place, virtual interactions are not constrained in the same way that real interactions are: to an extent, they may be completely

* 0-7803-8566-7/04/$20.00 © 2004 IEEE.
arbitrary. Thus, a programmer can sit down and, within the limits of the Turing-machine, do whatever he or she wants. The laws of physics don't apply.

To reconcile the virtual machine with its physical implementation — which is forced to undergo real interaction — we can consider what we call the complementary function: literally, this complements the virtual machine, allowing it to perform while the whole thing obeys the laws of physics. In practice, the complementary function is the combination of sensors, actuators, heat-sinks, casings, and so forth: everything that a symbolic or information-based machine lacks.

Our contention is that the better a system is coupled to the environment — the more the changes in the outer world change the structure within — the better will be the adaptation and learning.

3 Benefits of the approach

3.1 Understanding approximation

In our analysis (in [3]) one of the key results was the potential advantage of analogue technologies over their digital (symbolic) counterparts for learning and discrimination in intelligent machines. This is because of the necessary elimination (filtration) of potentially-useful information before it can be assessed for relevance by the intelligence. This process is known as analogue to digital (a/d) conversion. Where a problem is well-understood, such conversion poses few difficulties: engineers simply have to consider what level of approximation will produce results of the desired accuracy.

However, the approximation process in real to virtual conversion will — in poorly understood applications — cause an error whose effect cannot be predicted and therefore cannot be guaranteed to be within specific bounds. This is true whether this conversion is from the real numbers to the rational (continuous environment to Turing-machine input), or a large finite set to a smaller one (discrete-state environment to a strictly-limited-precision digital computer). One of these situations is where the function being performed by the embodied agent is chaotic: in this case, inputs may be arbitrarily close and outputs arbitrarily far. Another is where the error function changes over time in a way that is not known. Since the agent's function is not only determined by it's own learning function (the way it is changed by inputs given its current state) but by what it has experienced, the machine's whole lifetime would have to be predicted in advance in order for the error function to be known. Only in this case (and barring the function becoming chaotic at any point) could the precision required be set in advance with confidence that the error would be within a specified bound.

This issue is somewhat reminiscent of the halting problem, in that the only way to know what happens is to run the system. In this case, instead of trying to find whether or not the algorithm halts in a given time, the designer is trying to discover whether or not the error goes above a given threshold. Of course, it is not possible to know this unless an analogue machine is running at the same time for comparison.

3.2 Taking the system as a whole

Computational neuroscience (the study of biological neural structures using tools more commonly associated with computer science and engineering) often uses information theoretic approaches to understand the brain. In a paper like [1], for instance, Shannon's communication theory (see [26]) is used to improve our understanding of the vision system of a blowfly. The pattern of incoming light is considered to be a message from the environment. It travels through the system (modeled as a noisy channel) and, based on the total transformation that occurs, it can be calculated how much information is lost on the way.

This method is very helpful in that it allows us to think about individual neural systems at the microscopic level. As we look at them and see structures that encode, and thus retain, information well, we can learn a great deal about why evolution has designed them this way. In areas of the brain that are chiefly concerned with light, for instance, it is not at all surprising to find structures that, from an information-theoretic standpoint, are particularly good at handling light signals.

However, the brain is not a well-insulated, serial electronic, system, and the problems that it solves are not well-insulated either. For instance, mutual information (see e.g. pp. 40-41 of [10]) often exists between several different sensory modalities, and the information coming into the system through a single channel may have more than one cause. Related to the latter phenomenon, independent component analysis (ICA, see, e.g. [8]) is a well-known means of separating different signals traveling down a single channel into the constituent sound sources. It generally requires a number of different sensors in order to work (say a microphone per source in the audio case). Essentially, ICA is a more thorough version of principle component analysis which works through correlation of the incoming signals. In the case of multi-sensory information, there are artificial systems (based on mutual information, see, e.g. [4]) that can attribute a single cause for information coming through different channels [9]: in this case, audio and visual.

Given these two abilities, both attributed to neural systems, a third naturally arises that has yet to be fully explored: the ability to separate information from several different sensor modalities travelling down channels together. In other words, imagine that the neural response of the eye is somewhat modulated by pressure waves from sound, and the response of the skin and ears are modulated by temperature, and so forth. Each of the minor signals (those that are not of primary interest for a given sensor) would normally be seen as corrupting noise. However, there is no reason in principle why these so-
called noise components cannot be extracted out: that this noise might eventually be considered useful signal.

Thus, communication theory — though helping us to understand the way various structures work (and so helping us to think about how to design artificial versions) — limits us to a relatively narrow view of the functionality. We can think, for instance, of the visual system as purely dealing with incoming light as a kind of first-order approximation. Combining this with issues related to the other senses might be considered a second-order approximation. Thus, the dangers inherent in this type of analysis to define global properties of the brain are clear: more kinds of information may be processed than we realize. What we label 'noise' may, in fact, be information added, not taken away.

This is an argument that suggests that the whole will always be greater than the sum of its parts. Not because the analysis of any of the parts are incorrect, but because they cannot all be combined into a single model: in simple terms, the number of simultaneous equations would make the problem intractable. Such limitations are on the model, however, and not on the physical brain itself. This would be true whether the brain were real or artificial.

### 3.3 Providing footholds for evolution

The complex structures that exist in biology can, as discussed in the last section, be considered in information-theoretic terms. But how did these structures evolve? Mammals with sophisticated light or sound-sensing organs evolved from single-celled organisms. We know that this happened through evolution. But how does a sense organ evolve when there wasn't one there before? By considering the entire physical object as a sensor, just by virtue of the fact that it obeys the laws of physics, this model allows nominal sensor effects caused by physical structures to be exploited.

This can be thought of in two ways. First, you can consider the effect in a single animal. Physical computation, for instance, explains the ability of deaf people to hear through their bodies as the Scottish percussionist Evelyn Glennie has learned to do [11]:

"Evelyn spent a lot of time when she was young (with the help of Ron Forbes her percussion teacher at school) refining her ability to detect vibrations. She would stand with her hands against the classroom wall while Ron played notes on the timpani (timpani produce a lot of vibrations). Eventually Evelyn managed to distinguish the rough pitch of notes by associating where on her body she felt the sound with the sense of perfect pitch she had before losing her hearing. The low sounds she feels mainly in her legs and feet and high sounds might be particular places on her face, neck and chest."

Thus, physical features that are not designed as sensors can nevertheless be exploited as such. This is only possible, because — in the physical computation model — no explicit measurement path is required for these non-sensors. If it were, in an engineered system there would almost certainly be no such path. The kinds of vibrations involved in hearing, even at low frequency, would be unlikely to be considered of primary importance in an information-theoretic analysis of a leg: they would be considered too marginal (in terms of purpose) and too small (in terms of signal) to be worth collecting. Nevertheless, the brain can make use of the information that the legs provide as hearing organs: and in Glennie's case it actually does.

Such small effects can also be exploited by changing the entire morphology of a species. For instance, there is much debate about the reason for the shape of the hammerhead shark [19]. Unlike an ordinary shark, this animal has an oblong-shaped head much wider than its body — wide ends at the front and neck — with the eyes on the front two corners of the oblong. There are various reasons why biologists believe that this shape evolved. First, there is an obvious sensory advantage in that binocular vision (depth perception) improves with the separation of the eyes. Another explanation is that this head shape has hydrodynamic advantages including increased stability while turning. Finally, sharks have electrosensory pores that assist them in catching prey. The wider head means there can be a greater number of these pores for the same density.

With the physical computation model, evolution could select for any of these properties (i.e. they would supply some kind of advantage that would allow the animal to survive and procreate better than its neighbors) because any physical changes caused by mutation, no matter how small, or how seemingly unrelated to either sensing or locomotion, could be exploited. For instance, a slight widening of the head produced by evolutionary accident, could have improved the survival of the hammerhead shark precisely because all three abilities (electro-sensing, depth perception, dynamical stability) were enhanced. Similarly, we can imagine that a hair-like structure — one that happens to vibrate at a frequency of interest and happens to be located near some kind of nerve ending — could eventually evolve into a hearing organ.

Another example of this might be the vision system of the sunfish [7]. This creature lives in areas where the water is relatively cloudy, causing much diffusion. However, the morphology of the eyes allows polarization-difference imaging: where images are acquired of each polarization (due to the shape and orientation of the photoreceptor anatomy) and then subtracted from each other. This removes most of the diffuse light, leaving a useful image for the fish to interpret. Again, in terms of evolution, the physical computation here is crucial.

From an engineering perspective, development of the physical computation approach could eventually allow the
use of genetic algorithms (GAs) to select for success in a much more powerful way. Though evolutionary algorithms initially focused software agents in artificial environments [20], the idea of engineering hardware this way has been around for some time: initially working with simulated physics [14]. Later, from the same group, Thompson [29] showed that small hardware effects can be exploited by GAs to solve problems in ways that would not have been possible using conventional circuit design techniques or software-based design packages. Also, Sims [28] demonstrated that it is possible to engineer brain and body together, with physical and informational attributes evolving together in a competitive genetic selection process (he used a virtual physical model of the environment to do this). By combining all three approaches, we may be able to evolve embodied agents that can multi-task, using their physical attributes to best effect in terms of sensing, actuating and processing.

3.4 Processing the unmeasurable and computing the incomputable

If the most optimistic scenarios for analogue computation are applicable (which depends on answers, as yet unforthcoming, from physics), physical computation has advantages over Turing machines. Literally, this means that analogue could compute the incomputable, where computable is defined as computable on any discrete-state machine. Also, real-interaction-based physical computation does not require the making of measurements in the way that virtual computation does. Consequently, physical parameters that could never be known, even in principle, can nevertheless contribute to the physical processing taking place. In an application where sensitivity to the environment is an advantage, this is an important property.

It is important to stress at this point that this advantage cannot be easily harnessed (if at all) for conventional computing applications like word-processing or database management. The advantage is derived by two things that make it incompatible with conventional sensor-processor-actuator systems. First, the ambiguity of the sensors (the whole device is a sensor) means that all physical influences can be taken account, but their sources cannot all be separated. For instance: we are subject to gravity, and any small changes in its force on us will affect the way we behave. However, that does not mean we know about all the different masses that have combined to create this particular force. Such ambiguity — while acceptable for applications where the macroscopic physical behavior of a system is what's important — cannot supply the information required for conventional informational tasks.

3.5 The analogue shell and nested virtual machines

Chris Toumazou [30] has talked about the analogue shell of a device: the outer sensors and actuators that allow the inner digital electronics to interact with the outside world. The physical computational model fits well with this metaphor. Though Toumazou does not work on artificial intelligence, the systems that he designs — which include mobile phones and biomimetic sensors — fit well into a hierarchy that is suggested by the model. There are three obvious examples we can discuss: the digital virtual machine, the analogue virtual machine, and the analogue real machine. The latter should be well-understood by now as it has been the subject of much of the discussion so far. However, the other two cases are worth further consideration.

For virtual interaction, the first extra cost is that of shielding: in order to be selective about the information acquired (information implies selection) the object must be resistant to the other physical influences around it. Second, the object must interpret the information coming in through its sensors (and information it is, in this context) and assign it some kind of meaning (even if this is purely in the context of which bit comes before which) before it can act: this tends to mean dramatically reducing the types of information that can be considered and the amount of information that can travel through those channels. For instance, we might consider visual information (which is really a kind of map of our local electro-magnetic field) important, then reject certain wavelengths for consideration because their source, for instance, is ambiguous. Without the relevant knowledge the signal, or information, becomes meaningless. We can understand these shielding and interpretive functions as being part of the complimentary function.

In fact, the electronics industry is entirely structured on the basis of this duality. On the one hand, designers treat resistors, transistors, capacitors, etc., in terms of mathematical functions that they will perform on an expected (in terms of dynamic range, frequency, etc.) signal from a known source. Software engineers do not even consider the implementation: they simply assume that the high-level functions they need will be permitted on the machine they use. At the lowest level, a completely different set of designers look at the physics of the systems that will make both of these higher levels work efficiently. You can think of this as the analogue shell (complimentary function) with a first virtual layer of the designed electronics and the second virtual implementation of the software. These levels have clear demarcations.

What is potentially interesting about an all-analogue implementation is that there may be many levels of virtual computation rather than the two I've just described for digital. In an information-theoretic sense, we can think of many layers of analogue circuits, each one throwing away more information from the outside world in favour of those signals that are deemed (by virtue of, for example, neural network learning) important. It could be argued that the visual system, as described, for example, by Hubel and Wiesel [18] does something like this. Rather than encode all information, retinal processing
elements seem to break down images into lines of different orientations, and so forth. This concept of filtering is also used in pattern matching through Fourier-transform-based image correlation. Instead of comparing the images, elements of the image are filtered out and considered as a collection of elements. The more closely the feature collection of one image matches another, the more closely the images themselves are expected to match.

We can consider adaptive multi-level filtering as a kind of nested virtual machine: where an agent takes as much information into the system as is feasible given the sensors and then deciding what to throw away based on experience. Work by Geoff Hinton (see e.g. [17]) points to how this might be done: he devised a network that creates its own ideal set of features to discriminate between relevant objects. In a fully-adaptive system, not only the features (and how they are compared) might start by being unknown, but also the very question of what constitutes a relevant object. These might then be used as primitives for higher-level abstractions. In other words, as we penetrate deeper — past the analogue shell to the core — the computation looks increasingly symbolic and representational. Crucially, however, this is a much more flexible system than one that requires immediate (and irrevocable) analogue-to-digital conversion. And yet it is compatible with conventional artificial intelligence in that it allows for a pseudo-symbolic core to emerge. Thus, conventional algorithms that people have devised to explain intelligent behavior in people may be considered accurate: the question is how to implement them within a nested virtual system implemented on an analogue machine and without the usual interface of software.

3.6 Intelligence is more than thinking

The concept of the complimentary function may be helpful also when considering the mind-body problem. Our bodies not only provide our conscious mind (the intelligence function in this case) with sensors and actuators, but also the un/subconscious mind which is not normally considered in AI. Other bodily systems — the limbic system, the spinal chord, even basic organs like the heart and lungs — all contribute to our behavior to a greater or lesser extent. Using the physical computation model, we can see that what many people consider to be artificial intelligence, particularly purely software-based approaches, concerns only the virtual machine.

Striking a better balance may be said to be the concern of the neuromorphic engineers: their work often goes much further, blurring the interaction boundaries. Projects like Harrison and Koch's all-analogue fly vision system and robot controller [13], Hasslacher and Tildens' analogue walking/light sensitive robots based on the nervous systems of small animals [15,16], or Lewis's bipedal robots based on central pattern generators [21,22], all have in common that there cannot be said to be a clear line between sensor, processor, and actuator. Indeed, there cannot be said to be a clear line between software and hardware.

Pushing this to an extreme, one could argue that full understanding of real interaction, virtual interaction, and the complementary function might go some way to addressing Brooks' recent question about the relationship between matter and life [6]. In essence, we may just need to go further to try to understand a whole brain and body in order to understand how a creature works, rather than considering mind or brain processes alone. Considering this issue in the 1940s, Schrödinger suggested that the secret to the difference between what is and isn't alive lies in the nature of its reproduction and the coding of its construction in genetic code [25]. More recently, Pattee has suggested the same thing, and pointed to the importance of self-knowledge (in the sense of what is, and isn't, the self: see e.g. [24], and the importance of physical symbols (sets of physical states that essentially 'mean' the same thing). Since the integrity of both the genetic code and the self (whether at the level of single-celled organisms or animals with a large and complicated structure) is based on physics and chemistry, a more material, less abstract, understanding of these issues may be helpful.

4 The difficulties of the approach

Although it may provide us with new conceptual avenues down which to pursue embodied artificial intelligence, the adoption of this approach has three major problems associated with it.

4.1 Design and fabrication

Complicated physical systems are notoriously difficult to design and can be expensive to build. According to industry experts, too few analogue engineers are being trained [5]. There may be several reasons for this. First, the mathematics involved in analogue electronic design is extremely difficult. Second, even if a designer (particularly at the academic level) has the right analogue skills, he or she may have to wait a long time (weeks) for designs to come back from a VLSI foundry. Third, a relatively small design mistake or miscommunication could render the manufactured chip useless. Finally, the process is as expensive as it is time-consuming. A similar argument could be made for the work of mechanical engineers and optical engineers. By contrast, programming (software engineering) is relatively easy, inexpensive, and can be done by a single researcher without assistance or delay.

Though research progresses towards simplifying the design process of optical, mechanical and analogue systems (see, e.g., [12]) , the level of difficulty is currently still vast in comparison to the easy of sitting down to programme a computer.
4.2 The unpredictable black box

There is another problem that comes through working with any machine that has an interface to the real world and which is not well understood: unpredictable behavior. In a desk- or laptop computer this is not a problem because, in most circumstances, it poses no safety issues. In an embodied intelligent agent — whether one designed to have the same physical abilities as a human, or an autonomous aircraft or power plant — this is not the case. The larger and more potentially destructive a machine is, the more predictable we generally want it to be.

While any embodied AI is likely to be complicated and therefore difficult to predict, a neural-network-based analogue AI is likely to be much more so. This is because the behavior of the machine is determined by its history alone, and not by algorithms designed by software engineers. In a sense, this could make the machines more human, as people are also considered to be black boxes, conditioned by their experiences. On the other hand, it would not seem to be worth risking such an unpredictable approach for straightforward engineering applications or those in which it is easier to simply have a human operator in control.

4.3 Implementation, not explanation

The physical computational approach is intended to allow researchers to think about the most efficient possible solution to a problem in a given physical environment. However, like other approaches such as genetic algorithms (natural selection), it doesn't seek to explain why a particular solution is better than another. In fact, the parallel with natural selection is very close. Both it and physical computation explain the mechanism by which intelligent evolution takes place in an environment, not what form it should take. Thus, conventional models such as those based on information theory have two advantages. First, they can explain why a solution is a good one. Second, if they are sufficiently small problems that they can be adequately computed on digital hardware and with a reasonable analogue to digital conversion, such models are much more cheaply and easily implemented than the physical computational approach would be.

This ease of implementation, along with the ability to devise and then test hypotheses about the way intelligence works, makes the conventional approach very attractive. It is only where efficiency and sensitivity (either physical, informational or both) are crucial that the physical computational approach becomes worth the extra effort.

5 Conclusion

Embodied human-like artificial intelligence is a particularly good example of an application where first-order information theory would not seem to provide a sufficient model of what's going on. Likewise, looking at all possible uses of an information channel is not an efficient way of considering a system. A machine's embodied nature, encompassing its brain, means it has a myriad of ways of interacting with the world. Looking at a few of these will only tell a small part of the story. They may provide a helpful explanation, but will not be able to explain global behavior.

The physical computation model is useful for such demanding applications because it reminds us that — quantum mechanics aside — noise doesn't exist in the real world: everything is signal until we start to put an interpretation on what is important and what isn't. In order to build systems that can fully exploit their interactions with the environment, we must avoid restricting the information that they can take in: instead, we should allow them to develop, evolve, their own filters as their priorities emerge.

References


Appendix 3:

Sunny Bains

*Extending neuromorphic engineering beyond electronics*

Brain Inspired Cognitive Systems (conference)

Stirling, 29 August - 1 September, 2004
Extending neuromorphic engineering beyond electronics

Sunny Bains
Dept. of Electrical and Electronic Engineering
Imperial College London
and
Dept. of Design and Innovation
The Open University
Tel: +44 20 7594-6341
Fax: +44 20 7823-8125 (Notify by e-mail)
E-mail: sunny@sunnybains.com
http://www.sunnybains.com

Topics: analogue electronics, neuromorphic engineering, opto-electronics, optical interconnects, mechanics, embodied artificial intelligence.

ABSTRACT
One of the advantages of the neuromorphic approach is energy efficiency, which comes from the exploitation of the physical behaviour of electronic devices. Taking the intrinsic efficiency of physics as a guiding principle, we can extend neuromorphic engineering beyond electronics to other technologies including optical, mechanical, and chemical. In this paper we consider the role that some of these other technologies may have to play in this area, describe some of the work that has already been done, and suggest some advantages of pursuing what we call a physical computational approach to AI.

1. Introduction
This paper considers how the concept of neuromorphic engineering may usefully be broadened to a much wider range of technologies than just electronics. In particular, we discuss how it may be more efficient to allow technologies that are normally considered part of sensing or actuation sub-systems to be considered part of the intelligence of the machine and therefore better integrated.

First we consider why it may make sense to consider this hardware- rather than software-oriented approach to artificial intelligence. Then, in section 3, we look at some lessons from neuromorphic engineering: the underlying challenge of building brain-like systems is dealt with in part 4. In sections 5 and 6 we consider two areas where integrating optics with our systems may be particularly useful: increasing synaptic interconnectivity and neural complexity. In section 7 we look at the issue of analogue-to-digital conversion, the advantages of avoiding it, and discuss robotic implementations where some processing is performed in the body (rather than it all being done in the brain. In addition, we briefly consider the advantage of avoiding digitization when it may be necessary to fuse interdependent sensory information. In section 8 we give an example of how optics and electronics can be used more powerfully when properly integrated, and discuss the general implications of this, after which we conclude.

2. Embodied AI as hardware problem
We are concerned with the design of hardware that will implement embodied artificial intelligence, ideally to the point that a machine will be able to react to the environment in ways that make it appear to be as intelligent as a human. By starting with hardware, rather than cognitive models or connectionist networks, we do not intend to imply that software is not important. Rather, we want to show that—with a task as challenging as animal- or human-like AI—hardware may represent a bottleneck that we cannot afford to ignore.

We can break the problem of intelligent interaction down into a process of sensing the outside world, learning and making decisions based on the information acquired, and then actuating to change the world as necessary. If we do this then some of the potential problems become apparent. Are we sensing the world sufficiently (in terms of number, type, and sensitivity of sensors)? Do we have enough computing power to process this information appropriately in order to make decisions about what to do next? Finally, can our actuators implement these decisions sufficiently well? Do they have high enough speed, accuracy, and degrees of freedom to make the appropriate action? Lastly, is the latency of the entire system (from sensor to actuator) sufficiently low that circumstances will not have changed dramatically (at the appropriate macroscopic physical level) by the time actuation takes place.

There are additional engineering constraints: power, size, and heat dissipation among them. We are unlikely to want to build an android that requires its own power station to operate. Or that is the size of a tall building. Or that is too heavy to walk on the floors of our homes and offices. Or that sings our hair if we get within ten feet.

From a hardware point of view we can see that there are sufficient issues to attend to without needing a full knowledge of the algorithms to be run. Further, hardware constraints might lead us to prefer some approaches to AI over others. The application constrains the hardware, then the hardware (may) constrain the software.

3. Neuromorphic engineering
Carver Mead is best known to many of us for pioneering a new way of exploiting analogue electronics. His approach came from an interest in implementing neural circuits in order to process sensory information. In a book that has come to define the field of neuromorphic engineering (Mead, 1989), Mead showed that brains are about a billion times more power efficient than conventional computers. Though part of this could be attributed to the inefficiency of the individual transistors, etc., he believed that a performance improvement of six orders of magnitude was possible based on two things: using less wire and more local interactions; and exploiting, not suppressing, the intrinsic physics (i.e. analogue nature) of the device.

The first of these two strategies is an architectural issue that is helpful for any computer whether digital or analogue. In an electronic system, the length of wire between one component and another acts as a power drain. Component A only really wants to charge up component B: that's all that's necessary for the computation. But to do that, it has to charge up all of the wire between the two components as well. The extra power required to do this is fulfilling no computational purpose, it is simply an architectural overhead. Though designers do spend a lot of time trying to keep the amount of wire to a minimum, the kinds of processes that we run on digital computers (and the gate layouts that make them possible) do not lend themselves to the ideal architecture: one in which devices only have to 'talk' to near neighbours. The opposite is true of neural networks which also have major advantages for sensor fusion, see e.g. (Klein, 1999). By changing to this more efficient interconnection scheme, a 100-fold power...
efficiency gain can be made.

However, the advantage of moving from a digital to an analogue neural system is potentially even greater. Mead showed that computation could be performed up to 10,000 times more efficiently (in terms of power) by not throwing away the true functionality of the electronics used. In other words, by using them in a more analogue way. For instance, take a memory cell. Intrinsically, there is nothing about such a circuit that forces it to be either 'full' or 'empty' (to contain logic 1 or logic 0); we simply choose to interpret the information that way when we read it out. Because of this choice, we are forced to use an extra cell for each bit of information we want to store. If we allowed the charge within the cell to vary continuously (or at least to be able to adopt different 'grey levels'), then we could store significantly more information in a much smaller space. In addition, we would need to use less energy to both store and read out the information (because we would only have to go to one location to get it).

The same kind of analysis is true of the information processing side. The electronic circuits we use in computing have interesting, and often useful, responses. For instance, they can perform multiplications or manipulate a signal differently based on its strength or timing. Some of these features are, says Mead, very similar to some of those we see in the brain's processing devices. But we do not really make use of these as computational primitives (low-level processes on which higher ones can be built). Instead, we reduce problems to AND, OR, NOT, etc., and force our hardware to give us the minimalist answer to these questions (1 or 0) we require.

From a power-consumption point of view, it is important to note that the forcing required to achieve this involves driving the electronics in a particularly inefficient mode. More energy is consumed for an incoming signal to trigger a sharp non-linear (thresholded) response, than a sub-threshold (closer to linear) response.

In effect, we are reducing the operations that each component can perform, and increasing the amount of energy it takes. Of course, we know why this approach is taken. If we need a precise solution, we would not want to use an amplifier to multiply one number by another. Depending on the noise and the extent to which the device varies from its specifications, we would be likely to get a different answer every time. Thus we string together many gates, each consisting of many different components, to perform a digital multiplication. We get a precise answer, but at a cost in terms of efficiency.

If we have an application where we don't need that kind of precision in the first place, then this approach makes no sense: in this case we are throwing away the useful physics of the devices themselves, recreating the same functions using logic gates, and have nothing to show for it at the end. In such applications, analogue computers can make a real difference in terms of power efficiency without sacrificing functionality.

An illustration of this point is the cellular neural network (CNN) invented by Leon Chua (e.g. Chua, 1998), and now being used as an image-processing tool in the vision community. The CNN is a device that is digitally programmable but can perform complicated non-linear operations during the analogue transient: the 'switching' time to go from one stable state to the next. It is an apt demonstration of Mead's point. Not only is the device far lower-power than the equivalent image-processor, but it is also up to orders of magnitudes faster (depending on the algorithms implemented).

It may seem redundant to cover this ground in a conference aimed at neuromorphic engineers, but it will be helpful to have them uppermost in our minds as we proceed.

4. The challenge of building brain-like systems
For many of us, the future of AI will involve, in one way or other, the building of artificial brains. The human brain has of the order of $10^{10}$ neural processors, linked by as many as $10^{15}$ synapses (Churchland et al., 1992), all contained in an object about the size and topology of a folded pizza (analogy courtesy of Christof Koch, California Institute of Technology). If each of these parallel processors were performing just one logic operation per emission of a neural spike, duplicating this system in digital electronics would be formidable challenge. In fact, according to Yaser Abu-Mostafa, it would be practically impossible today because of the lack of connectivity between neurons (Abu-Mostafa, 1988; Abu-Mostafa, 1988). In his papers (and echoed in an appendix he wrote for Carver Mead's book (Abu-Mostafa, 1989)), he argues that the ability of a local-learning neural network (which would hold for the biological case) is limited by the number of connections between the neurons. This, he says, cannot be made up by simply using a larger network.

If he is right, and biological neural networks do indeed fall into this class, then this could represent a major bottleneck. We have three options. The first is to devise a learning rule that allows us to produce the same discrimination ability with a modest level of interconnectivity—of the order of 1-10 connections per neuron rather than the average of 1000 in biological systems—by using some kind of size/interconnection trade-off. The second is to find some way to improve connectivity. Finall, the third option is to give up on the neuromorphic approach all together. The first and third of these options may be the same. According to Abu-Mustafa, the local learning rule was considered precisely because it was biologically plausible. Departing from it (necessary to overcome Abu-Mustafa's limit) by definition changes the nature of the system being designed. In any case, connectivity is an issue that has had to be addressed for many connectionist (neural-network-based) and distributed-computing applications.

5. Connectivity as a hardware problem
David Miller (Miller, 1989, 1997; Miller et al., 1997) has written extensive critiques on the problems of on-chip electronic interconnections. The problem can be expressed simply: the longer an electronic interconnect is, the higher its capacitance. This means that more energy is required to get a signal from one end to the other and, since circuits have to be designed with the worst case scenario in mind, the performance of the entire system ends up being determined by the longest link. As a result, designers try to keep interconnects as short as possible; nearest-neighbour interconnects are ideal. Similarly, the more interconnects a chip has (of whatever length) the more power it must produce to charge them all simultaneously. This is particularly true when broadcasting a signal: sending it through all its interconnects at once. Thus the number of interconnects becomes a limiting factor. Since broadcasting does seem to be an important part of some neural functioning (see e.g. (Abu-Mostafa, 1988)), these scaling issues must be considered.

When information must be exchanged between one
chip and another, there is a further problem: a bottleneck is caused by the fact that electronic die are essentially two-dimensional. If a square chip is length \( x \) on a side, then the area (and number of processors accommodated) varies as \( x^2 \), while the number of interconnects (pins along the edges of the chips) varies as \( 4x \). As silicon wafers get larger and feature sizes get smaller, this problem gets worse (even if the size of pins scales down too).

Finally, there is the problem of crosstalk. Interconnects must be electrically shielded from each other, otherwise the electric field will create false signals (noise) in neighbouring wires. This leakage also represents power dissipation. The need for shielding represents a limit on the (minimum) volume required for an electrical interconnect.

Inventive ways have been found to get around these problems for experimental systems. The best known electronic method, known as address-event representation (AER, (Boahen, 2000; Mahowald, 1992)) essentially gets around the broadcasting problem by using a time-multiplexed, pulsed network. In this asynchronous (analog) system, individual pixels request access to the bus when active. For example, Eugenio Culurciello (Culurciello et al., 2001) built a system where pixels in an imaging array, via artificial neurons, emitted spikes at a frequency proportional to the light intensity. Since AER is a responsive network—access to the communications bus is granted on request—the available bandwidth is allocated according to need. Also, because the spike time is small with reference to the inter-spike-interval time for a given neuron, delay in getting access to the bus does not significantly change the pulse-coded signal.

Unfortunately, though it represents an ingenious and efficient use of infrastructure, AER does not change the fact that there is a serious scaling problem. It was designed to scale well (which it does) for increasing numbers of processing elements on a single array: For \( N \) elements in an array to connect to the same number in another array, only \( 1 + \log N \) wires are needed. If \( M \) such arrays have to be connected with each other, then \( M (1 + \log N) \) wires would have to be in place. With current technology, only relatively small numbers of chips can be connected in this way. In addition, AER is not designed for on-chip communication, so there is a trade-off between making the arrays bigger (for fewer chips in the network) and local connectivity.

There are other techniques under development, some of which are likely to be compatible with AER. Most of these would fall into the broad class of optical interconnects. Unlike wires, optical signals don’t have crosstalk (unless handled poorly at the detector), and so can coexist in the same volume. This comes from the intrinsic physical fact that photons neither repel nor attract each other, nor do they interfere with each other in any way. (Photons, when split, do interfere with themselves, but that is another issue, and not relevant here).

The most obvious example of this is the wavelength-division-multiplexed fibre-optical link (Miki et al., 1978) where hundreds of closely-spaced wavelength channels can propagate down the same optical waveguide and be differentiated at the other end without their signals becoming mixed. As the light sources, non-linear materials, detectors, etc., for photonic systems become cheaper and more available, increasingly ambitious systems are being built. For example, Nan Jokerst built a substrate-guided or planar-optical system at Georgia Institute of Technology (Jokerst et al., 2000). Essentially, this gives the light freedom to move in \( 2^{\frac{1}{2}} \) dimensions: the two dimensions of the plane of the interconnect, plus up and down (into devices attached to either side of the plane). By including emitters and detectors on the top of circuits, signals can be either actively or passively routed to other locations (potentially many other locations) on the chip or board.

Among many other notable examples is NEC’s optical backplane network at its Laboratories in Princeton (Araki et al., 1996). This was aimed at providing slow reconfigurability for massively-parallel computer systems and used free-space optics (optics in air/gas) rather than guided-wave optics. Based on an analysis of distributed computing systems, the NEC network was designed to use an electronic crossbar for local interconnections on each board, and had vertical-cavity surface-emitting lasers (VCSELs) emit signals to travel between boards. Each board needed a VCSEL and detector for each other board it has to talk to, and the address is specified by the geometrical location of the VCSEL in the array. This kind of scaling may seem similar to the one that posed such a problem in AER. However, optical paths can cross without affecting each other and, since they do not have problems with heat dissipation, they can be arranged in close-packed arrays, with the area of the array scaling linearly with the total number of interconnected boards.

A more radical solution, using both fibres and free-space, was designed by Ed Frietman at the University of Delft (Frietman, 1995). Rather than producing a network that on some level requires message passing, like the NEC and AER schemes, this system was based on every processor talking to every other processor. This works by each sending out information through an array of light-emitting diodes, one for each bit. This light is then captured by a polymer fibre optic array and connected to a central node known as the Kaleidoscope. The fibres are organized so that their relative positions at the output are the same as at the input: effectively making the data into a 2D image. In the Kaleidoscope, the fibre bundles are tiled to make one large 2D image, the size of which is dependent on the number of bits output by each processor and the number of processors in the array. Using a faceted mirror and lens system, the light from the entire image is broadcast onto separate locations for each processor, coupled into a second fibre bundle, and then received so that the processor can access the data electronically.

It should be noted that not all current strategies to increase connectivity are optical. Irvine Sensors invented a new mechanism designed to be compatible with its chip-stacking technology (Carson, 2000). The stacking technique involved producing essentially conventional chips and removing the substrate (wafer) through rubbing. Apart from allowing more devices to fit into the same volume, this also provides a new opportunity for interconnection. To capitalize on this they invented the three-dimensional field-effect transistor. This device is constructed as two parts on two separate chips: once they have been bonded together in a chip stack, they operate as a single transistor. This extends the concept of a nearest neighbour to three dimensions rather than two: as it is in the brain.

Though this work is not optical, it is interesting to note that Irvine sensors is a company with a history of supporting research into optical interconnects (some of which has recently been published, (Li et al., 2002). Further, in the Carson’s speculation about building brain-sized systems, he has always assumed that it will take many different stacked modules to produce the required power. Interconnection between modules could therefore be optical.
6. Capturing the complexity of neuron interactions

From our discussion of Carver Mead's work earlier, we know that the replication of simple analogue functions is more efficiently performed using sub-threshold electronics. However, there is much evidence that real neurons are more complex than even Mead's circuits (see e.g. (Churchland et al., 1992) or (Wehr et al., 1996) for a more specific example). Analysis of one of the best-known neural circuits, the Hodgkin/Huxley model of the squid axon (Hodgkin et al., 1952), suggests that it performs in a mathematically complex way (see e.g. (Arcas et al., 2000)). The excitable membrane of the squid's axon becomes increasingly depolarised due to the incoming signals (spikes or pulses of ions) until it reaches a threshold potential. At this point it abruptly inverts, producing the neuronal action potential and, thus, the output signal. Afterwards the membrane undergoes a period of no response, followed by another slow build-up. As a result, a pulse arriving immediately after firing will have a completely different effect on the output of the neuron than a pulse arriving immediately before. Thus, spike timing is critical.

Nabil Farhat has done significant work in this area, both from the dynamics perspective (Farhat et al., 1996) and the hardware perspective (Farhat et al., 1995). He claims that the real complexity comes from combining this behaviour with the processing carried out by the neuron's receptors in the dendritic tree. Correlations, caused by synchronicity in the incoming signals, cause a periodic modulation at the excitable membrane. This gives rise to complex, ordered patterns of firing that are phase-locked to the periodic modulation, and to disordered (chaotic) firing that depends on the amplitude and frequency of the modulation. In this way the neuron detects coherence or meaning in arriving spike trains and encodes this information in its own output.

Here, an analogy can be made here between spike timing and optical phase. In photography, only the combined intensity of light arriving at the film is recorded: the phase information (encoding distance, direction) is thrown away resulting in a flat image. In holographic imaging, the phases of incoming rays are allowed to interact through interference, and the result of that interaction is recorded (Hecht, 1987). This is what encodes the image’s third dimension, depth, and increases the amount of information that can be stored and retrieved (Mok et al., 1992). In fact, researchers have exploited optical interference to make associative memories (e.g. (Farhat et al., 1985), based on optical correlators (Casasent et al., 1976; Juday et al., 1987) combined with holographic data storage (Psaltis et al., 1990). Such devices have the advantage of both of high density—through efficient use of material—and content addressability: possible because the geometrical/topological integrity of images is retained in the way they are stored.

Farhat built both simulations and actual analogue models of neurons that could take timing into account in this way and found that they produced a bifurcated output: they depended on only small changes in input frequency and phase, periodic, m- and quasi-periodic, and chaotic firing patterns were all observed. Further, Farhat designed a system that could take advantage of the high-connectivity available through optical interconnects, the complexity of analogue neurons, and the unusual properties of electron-trapping materials (ETMs): the latter added a further layer of (biologically-plausible) nonlinear interaction to the network.

It is worth understanding how the additional complexity is achieved. ETMs are materials that contain two sets of impurities: one with an electron that is easily liberated (Eu2+) and another that provides a trap for it (Sm3+). On illumination with blue light, electrons are excited and either fall back to the Eu2+, producing orange fluorescence, or become trapped. On illumination with infrared, trapped electrons tunnel back to the Eu ions and then fall into its ground state, again producing orange light. Electron beams can also be used instead of the blue wavelength, increasing the complexity of the interaction. Normally these materials are used in a read-write series: where the IR reads out what the blue records. However, when IR and blue light illuminate the ETM simultaneously, the dynamics become complex: especially if one or both beams are changing with time. As a result, Farhat determined that the ETMs could provide input and output (dendritic and synaptic) weights, nonlinearly coupling bifurcation neurons together.

In (Farhat, 1998) Farhat showed that, in simulation at least, the bifurcation neurons formed nonlinearly interacting phase-locked netlets or neuronal assemblies with external input. The chaotic periods served as noise, and assisted neural functioning through a kind of stimulated annealing process: it would help the netlet to converge by popping it out of local minima. According to Farhat, this behaviour was very robust at the netlet level, even if the reactions of particular neurons within it were imprecise. Further, he says that this behaviour seems to mimic that of cortical neurons and could be potentially useful in creating intelligent machines. Experiments with similar networks (Farhat, 1997) have shown that they can be used to accurately discriminate between objects. Similar conclusions about noise have been reached by researchers at Boston University (Mar et al., 1999).

The fact that complex (analogue) neural behaviour exists in nature does not prove that it is functional, but Abu-Mostafa argues that it is. In his appendix to Mead's book (Abu-Mostafa, 1989), he shows that not only does using analogue neurons with more than two inputs reduce the number of neurons required (to replace N K-input analogue neurons, a minimum of N K^2 binary neurons would be required), but there is no guarantee that such binary network would do the job.

That said, there is no reason in principle to argue that neuron complexity cannot be traded-off against speed, network size, etc.. However, in practice (and here we are concerned with the engineering aspects of systems), though imperfect, biology is generally very efficient. For example, the use of ATP (adenosine triphosphate) as a fuel for motor proteins (such as kinesin and myosin) is highly efficient (Howard, 2001). It has almost no thermal by-product, turning chemical energy almost entirely into mechanical energy. By over-simplifying neural processes and ignoring the complexity of biological exemplars, we risk turning human-like embodied AI from a problem that could be solved practically (implemented in a machine the size of a man) into one that cannot (where the machine would have to be the size of a planet).

7. Hybrid system design and A/D conversion

Most work in embodied artificial intelligence (see e.g. (Johnson et al., 1995)) has a conventional structure: sensors receive analogue information from the outside world, turn it into bits, and send it to be processed by a digital computer. Once the processor has decided what to do, the decision is passed to a controller that turns this information into the
appropriate (often analogue) driving signal for the actuators. Such machines can be defined as hybrid—both analogue and digital—and their structure as conventional: the A/D and D/A conversion stages necessitate the isolation of each of the sensor, processor, and actuation stages from each other.

It is not always necessary to design systems in this way: instead, all three stages can be merged into a single physical system. Such an approach has two main advantages. The most obvious is in terms of speed and power consumption. This fact is increasingly being recognised by roboticists. For instance Matthew Williamson, who worked on Rodney Brooks Cog project, was charged with engineering the robot's limbs in such a way as to ease both the information-processing and energy burden they represented for the machine as a whole. A mechanical engineer by training, he designed Cog's arms and wrists so that they were compliant (Williamson, 1995) and could respond mechanically to changes in the environment rather than purely through conventional sensor-processor-actuator loops. The result was not only a more mechanically-efficient and natural-looking movement, but considerably lower computational overheads.

Others interested in exploiting a balance between information processing and a machine's physical dynamics (in this case, mechanical) computation include Lungarella and Berthouze (Lungarella et al., 2002). They have looked at how temporarily restricting the degrees of freedom of a mechanical system can improve a robot's ability to learn how to manipulate it. In another recent paper, Pfeiffer gives numerous examples of the importance of mechanical design to machine intelligence (Pfeiffer, 2002). Going further back, biologically-inspired roboticists have looked at how cats survive falls from tall buildings through a mechanical process of turning, and through using their legs as a buffer between critical systems (the body and brain) and the ground (Cameron et al., 1991).

Another advantage of avoiding the conventional setup is that it is not necessary to determine what resolution of A/D/A conversion is necessary. Conventionally—in the case of machine vision, for instance—an engineer has consider how many grey levels are necessary to implement a particular task (such as floor-following (Horswill, 1993), defect detection (Davies, 1990), or number recognition (Hinton, 2000) in a particular set of conditions (eg. over a given range of light levels, object orientations, distances, etc.) in order to specify the hardware. Though machine vision is still at a primitive stage, the reverse-engineering process for choosing sensor sensitivity, resolution, and dynamic range is generally well understood and is used for most applications.

However, because one of the prerequisites for such reverse engineering—a clear specification—is unavailable for our application, the interface between analogue and digital layers becomes a problem. This fact is intrinsic any basic definition of human-like AI: the machine must be free to adapt to its environment based what's important to its survival. This suggests that the machine should be free to use the dynamic range of its sensors as it sees fit, without an artificial A/D conversion limit. If the maximum bit resolution is specified in advance, then information that came in through the sensors may be withheld from the processor. As this information is unavailable for scrutiny then the embodied agent will never be able to learn whether it was important or not.

In humans and animals, sensors output their information as asynchronous spike trains; the timing between spikes can vary continuously and is not regulated by a clock. This freedom from being forced into discrete values can be useful. For some problems, non-linear greyscales (such as logarithmic, see Weber-Fechner law, e.g. (Walker, 1995)) are preferable to linear: particular light levels may more important than others and require more discrimination, others less. Further, if the scale can be changed for different applications or environmental conditions, then the whatever information is available may be exploited to its maximum potential.

Perhaps the clearest example of where A/D/A conversion can be a problem is in systems where feedback is important. In electronics, positive and negative feedback are used to either minimize or maximize small changes in an input, thus making a circuit either stable to small changes or extremely sensitive to them (Young, 1988). A nice example can be drawn from opto-electronics (Dupertuis et al., 2000) where a semiconductor optical amplifier (SOA) can be made several times faster without reducing gain or increasing current through the injection of a continuous-wave light beam. The technique exploits the transparency point of a photonic device—where a wavelength is absorbed and emitted equally—using the incoming light to boost the production of carriers when they are most needed. The problem is that, when the incoming signal at one wavelength is amplified by the SOA, there is a decrease in the number of charge carriers available to produce gain. Electronically it takes a long time to replenish these. However, the decrease in charge carriers has a secondary effect: it shifts the transparency point of the material. The continuous-wave beam, which is at a different wavelength, is at this transparency point, having almost no effect on the device before the signal arrives. While the signal is present and amplification takes place, however, the resulting drop in charge carriers pushes the injection wavelength into the device's absorbing region, and the absorption, in turn, quickly produces the needed charge carriers.

Another example, particularly relevant for our application, is local inhibition in the retina (for a brief review and recent developments, see (Roska et al., 2000). Here, small differences in the intensity of light received are amplified so that, amongst nearest neighbors, only the brightest pixel fires. This principle has been incorporated into neuromorphic systems called winner-take-all networks (Lazzaro et al., 1989) and these have been used in navigation and sensing (e.g. (Indiveri et al., 1996)). Part of their advantage comes from their analogue nature, which means it is almost impossible that two incoming signals can appear to have the same intensity. Even the smallest difference can be leveraged to produce a clear difference. If digitized, this would not be the case.

Another area in which resolution issues can become important is sensor fusion: this is particularly because of the inter-dependence of sensor systems within a complex embodied intelligence such as a human. In essence, the sensors perform an application that is more than implementing a particular sense. Specifically, the eyes must not only supply sufficient information to allow, for instance, visual pattern recognition to take place, but it must supply sufficient information to guide and supplement (for example) locomotion or audition. Further, it must supply this information in an appropriate form.

To give a more specific example, in the spinal cords of many vertebrates (Cohen et al., 1992), sensory information is directly incorporated into the gait of the animal through a process of entrainment. Cohen and colleagues have modelled the interaction with visual...
feedback both theoretically and experimentally. The spinal chord has its own driving frequency, the source of which has been modelled as a coupling between adjacent oscillator sections (such as the vertebrae). The coupling takes place through the neural integration of spike trains, and so can be manipulated by the addition of extra spikes.

The word ‘entrainment’ refers to the modulation of one signal by another (similar) oscillation, with the output both frequency- and phase-locked to the latter. This kind of entrainment does occur with some sensory input: particularly tactile sensors directly connected to the locomoting limbs. However, in the visual case, the signals are not of similar frequency. Instead, the integrating neuron reaches its threshold condition earlier when additional excitatory spikes are added, more slowly when the extra signal is inhibitory. In this way, the analogue signals from one sensory modality can directly interact with the behaviour of another system: without any explicit processing.

This is a crucial point. Studies of the brain in many different animals (see examples for human, monkey, and cat in (Foxe et al., 2000), (Dhamel et al., 1998), (Wallace et al., 1997), respectively) have shown that visual and auditory signals are not just processed by their own processor (or cortex), but that there are connections that leave each of these sensory systems very early in the chain of processing. Thus, the resolution required for visual applications is not relevant in providing the correct determination for those systems that share visual information. As in the spinal chord example, such systems are not necessarily making use of the results of conventional visual processes (like pattern recognition, object tracking etc.). Rather, they are processing the visual signals in their own ways for their own ends.

Because of the fact that the sharing systems—those directly exploiting sensory signals that are nevertheless outside their primary modality—are working with relatively raw (unprocessed) signals, they can potentially make use of any information that such signals contain. If the signal is digitised then, again, the information available has been artificially restricted. Once again we return to the problem that, in order to be able to set a resolution that is indistinguishable from the original analogue, then it is necessary to fully understand both the neural processes and the applications they are serving. Such an expectation would not be permitted given our definition of an embodied human-like artificial intelligence given previously.

8. An integrated approach
An excellent example of the advantages to be derived from taking a more integrated approach—in this case integrating optics and electronics—to designing intelligent systems is illustrated by CDM Optics (Boulder, CO) wavefront coding. This approach attempts to match the optical design to the detector electronics and application. In it, unexploited resolution is traded for increased depth of field and a slight increase in noise. The resolution may be unused, for instance, because the pixel separation is significantly greater than the focal-spot size, and can be exchanged for either a deeper image, or for one with a more relaxed focusing tolerance (see figure). One application of this is in color imaging: a single lens can be designed that focuses light over a broad bandwidth—like white light—without chromatic dispersion. The technique works by designing optics that have a large—but uniformly distributed—point-spread function (PSF). Images produced by such optics appear blurry but, because the blur is uniform, can easily be extracted by image processing: the PSF can be used as a kernel filter.

Lenses designed to produce the wavefront to be decoded generally look very different from conventional optics. In the case of the conformal IR-imaging system, the imaging side is conventional but the detector side has a cosine-form surface—one with three teardrop-shaped peaks circled around, and pointing toward, the center, [Kubala 2003]. This design not only takes account of the imaging characteristics that are required, but also of the image-processing overhead. Because increasing the kernel-filter size rapidly increases the processing time, the size had to be limited to 10Å: ~10 pixels. The ability of wavefront coding to accommodate such constraints is one of the things that makes it so powerful.

Examples such as this one—as well as many of the others in this paper—show how important it is that sensing, processing, and actuation as not treated as different design problems but part of the a single physical problem. We have developed a model of physical computation [Bains 2003] that seeks to show why this is the case, and have begun to assess the implications of this fact for engineering. One of the most basic results of this work is that information processing is just a slice of what any given physical object (even one built entirely for information processing) can do. Understanding the relationship between the virtual machine and physical machine is, we believe, the first step towards a more integrated approach to co-designing the brains and bodies of robots.

Conclusion
In this paper we have described a number of different projects all of which, we believe, point to the fact designing the most efficient embodied intelligent systems requires an integrated approach to robotics. Such an approach, which is consistent with neuromorphic engineering, moves beyond the consideration of brain (electronics) and body (optics, electronics, mechanics) as separate sub-systems. Instead, it seeks to combine them and integrate them fully, exploiting the processing ability of all system components regardless of their primary purpose.

References


Casasent, David and Demetri Psaltis (1976), Position, rotation, and
Appendix 4:

Sunny Bains

*Intelligence as Physical Computation*

Intelligence as Physical Computation

Sunny Bains

Intelligent and Interactive Systems, EEE Department, Imperial College London
Exhibition Road, London SW7 2BT, UK
Department of Design and Innovation, The Open University
Milton Keynes MK7 6AA, UK
sunny@sunnybains.com

Abstract

In this paper, we describe a model of physical computation that allows us to understand how embodied intelligent agents can simultaneously be considered to be objects operating under the laws of nature (or physics) and information processing devices. Using this type of analysis has two major advantages. First, it allows us to make concrete those issues relating to the relative merits of using analogue values or symbolic representations such as numbers: our analysis allows this longstanding point of contention in artificial intelligence to be transformed from a question of philosophy to one of physics. Second, it gives us a framework that should eventually allow us to produce design rules that will enable our artificially-intelligent (physical) agents to make better use of influences from the physical environment. As should be expected, for well-defined tasks, the results of this analysis are identical to Shannon’s information theory. However, for large, multi-functional systems with ill-defined roles, the new model provides a novel way of thinking about the complementarity of hardware and software.

1 Introduction

In this paper, we take a fresh look at what intelligence is, where it fundamentally comes from, and how this affects the way we think about building artificially-intelligent agents. Note that this model is exclusively directed at the goal of understanding embodied intelligence and, particularly, intelligence that is based in the natural or real (as opposed to mathematical) world. In this sense, though it applies to any real machine that can perform mathematics, the model is least useful for those whose goal is mathematical manipulation rather than the processing of signals from the outside world in order to produce intelligent actions.

This research has come about through a desire to bridge a gap between two existing fields. First is theoretical work that proves that analogue computation, in the form of recurrent analogue neural networks (RANN), can be super-Turing in nature (Siegelmann, 1999). Siegelmann demonstrated that, if allowed to take continuous rather than discrete weights, recurrent neural networks could perform functions that are theoretically impossible using Turing machines. The problem with this model is that it is entirely mathematical: in it, there is no interface to the real (physical) world, so assumptions she made (about continuity, for instance, or noise) have no clear way of being validated. Likewise,
Physical Computation

studies that look at how noise can degrade the abilities of such machines (Maass, 1997) do so from a theoretical rather than physical perspective.

The link between the theoretical and the physical is important because the brain looks very much like a RANN. If Siegelmann’s conclusions were applied to the brain, then Roger Penrose’s (1989) assertion that human-like intelligence could not be performed on Turing machines could be correct (though for the wrong reasons).

It is important, at this point, to distance the arguments made in this paper from the debate Penrose famously started concerning physics, artificial intelligence and computationalism. Using Gödelian arguments (Gödel, 1931), he pointed out that certain propositions are undecidable in Turing machines (as they are in all such mathematical systems) based on the axioms inherent in those systems. He also claimed that microtubules in the brain (Penrose, 1997) had quantum-mechanical properties that were both non-Turing-computable and potentially important to intelligence. These claims have generally been disputed from both computational and physical points of view. First, it has been argued that the specific Gödelian limitations are not, in fact, in conflict with human intellectual abilities (e.g. Laforte et al., 1998). Second, the timescales related to quantum decoherence in microtubules have been shown to be so different from those relevant to the brain that classical (rather than quantum) neural behaviour has been proposed as the more appropriate model (Tegmark, 2000).

Instead, this paper is more concerned with issues such as those succinctly outlined by Dreyfus in the 1970s (Dreyfus, 1972). In his book What computers still can’t do: A critique of artificial reason, Dreyfus explains why the formalization of a physical process is not the same as the process itself (Chapter 5). With this paper, we take a more engineering-based approach to his philosophical questions. We ask, if the behaviour of physical systems cannot be replicated using Turing machines, how can they be replicated?

Back to the technological gap that we are trying to bridge. On one side we have the work done in theoretical computer science (by researchers like Siegelmann), while on the other we have the field of neuromorphic engineering: where engineers try to structure their machines, often including the hardware, in a brain-like way. Carver Mead’s analysis of the power-efficiency of analogue computation (e.g. Mead, 1989), particularly for neural networks, has been extremely important in this regard. He demonstrated that by exploiting, rather than fighting against, the intrinsic physics of electronics, analogue circuits could be orders of magnitude more efficient than their digital counterparts. Leon Chua’s cellular neural network (e.g. Chua, 1998), a device that is digitally programmable but can perform complicated nonlinear operations during the analogue transient—the “switching” time to go from one stable state to the next—is an apt demonstration of Mead’s point. Not only is the device far lower-power than the equivalent image-processor, but it is also up to orders of magnitudes faster (depending on the algorithms implemented). Nabil Farhat’s optical implementations of biological neural models (e.g. Farhat, 1997 and Farhat and Wen, 1995) not only show the utility of analogue networks, but their potential complexity. From conceptually very simple optical and/or analogue circuits, he has obtained behavior that—with within a single system—varies between periodic, m-periodic (repetition of a pattern of m beats), psuedo-periodic (qualitatively periodic, but not strictly so quantitatively), and chaotic (with a fractal set as an attractor) output.

Though these researchers never made any explicit claim that their systems were computationally superior, Siegelmann’s model suggested they might be. Thus, it seemed logical to find an approach that would bridge the gap between the two. To date, Siegelmann and others (e.g. Blum et al., 1989) have worked with notions of super-Turing or hyper-computation: forms of computation that can perform functions theoretically impossible with conventional Turing machines, such as functions based on non-computable algebras.

http://www.aisb.org.uk
or that are non-recursive. In particular, they have concentrated on computation over the reals. This theoretical approach, though fruitful, does not tell us what is possible with real machines. Our new model, on the other hand, was designed with the following intentions: a) that it should have a clear interface with physics; b) that it have a clear application to artificial intelligence in the simulation-of-behaviour sense (as opposed to the solution-of-arbitrary-mathematical-problems sense); and, c) that it allow for an understanding of the differences and relationships among various different types of machine.

Note that, though the model presented here does all of the above—including providing an interface with physics (in that it can be interrogated using what we know of the physical laws)—physical interrogation is beyond the scope of this paper (but is at an advanced stage).

The paper is structured as follows. In Section 2 we discuss the Turing machine and why, whatever its comparative computational power, it is an inadequate model for the embodied intelligence task. In Section 3 we introduce the basis of the new model, showing some important variations on how it can work in Section 4. The basics of how the model can be used as an interface between the computational and physical paradigms are explained in Section 5, with a discussion of the model—with particular reference to neuromorphic engineering—in Section 6. Finally, in section 7, we will describe current avenues of this research that are beyond the scope of this paper, and future avenues that may be productive.

2 The Turing machine as a model of intelligence

When Turing wrote his original paper describing the thought experiment that became known as the Turing machine, he described it as a kind of automatic version of doing what people do when they manipulate numbers and symbols on paper in an intelligent way (Turing, 1936). The machine had an arbitrarily long tape from which it could read as many symbols (from a finite set) as needed, means to read and manipulate those symbols (the read/write head), a finite set of states where it could store information about what had gone before, and a set of rules that governed what it should do in the event of various combinations of input and state.

Critically, the input and output are explicitly constrained to be symbolic. Thus, the machine is unable to interact directly with the environment: it must always receive a set of symbolic inputs from sensors, which have to perform some type of transformation into the relevant symbol space (such as digitisation), and must produce its output as a set of control symbols that can then be used to affect the real world.

This fact is critical, because it means that there is no way that a Turing machine can perform any kind of natural behavioural intelligence—where it responds to some external force or signal by moving or changing state—it can only perform the intelligent manipulation of symbols. It is, therefore, not just wrong but essentially meaningless to speculate on the ability of Turing machines to be able to perform human-like intelligence tasks. In any real machine, but particularly those designed to interact with the environment, the outer shell (body, sensors, actuators) must be, by definition, entirely analogue. This is because the signals that they deal with (from the outside world) are analogous to the real physical values in question (or, more precisely, they are the real values). For the machine to work, at some stage after this outer-shell has been breached, an analogue-to-digital (a/d) conversion step must take place, thus allowing the digital computer to do whatever processing is required. The same, in reverse, is true for actuation.

Given this, every robot (and, to a lesser extent, every computer) is a hybrid machine:
part analogue, part digital. In the following sections, a crucial question in the design process of any machine, but particularly a machine intended to be tightly-coupled to its environment, is highlighted: the location of the boundary between these two parts. In particular, we believe the virtual interaction variation of the model gives some insight into this question though, as yet, it falls short of providing specific design rules.

Note that the above argument is not anti-computationalist in the traditional sense. Rather, the intention here is to point out that computationalism as it is normally discussed is moot. Since all robots must be hybrid machines (and, therefore, not necessarily constrained by the well-known limits of Turing computation), setting up the Turing machine as the only route to artificial intelligence is, in effect, setting up a straw man. From a theoretical perspective the question is whether the hybrid machine has the computational power to do the job. From an engineering perspective, the question is how such a machine can be designed to do the job most efficiently.

3 The physical computational model

Here we present a mathematical model of a potentially-intelligent, embodied, adaptive, physical system: one that includes mechanisms that can be interpreted as allowing learning via experience of the environment and action based on that experience. In Section 5 we consider the elements that are affected by physical constraints (i.e. the constraints of the real physical world as opposed to specific engineering constraints), but for now we simply lay out the mathematical model.

A system is here defined as an identifiable collection of connected elements. A system is said to be embodied if it occupies a definable volume and has a collective contiguous boundary. The matter, space and energy outside the boundaries of the embodied system are collectively called the environment.

A sensor is any part of the system that can be changed by physical influences from the environment. These, which include any or all forces, fields, energy, matter, etc. that may be impinging on the system, are collectively called the sensor input \((x \in X)\), even where no explicitly-defined sensors exist. Though represented by a single variable, the sensor input may in fact consist of many different sensor modalities (each influenced by a different type of force or energy).

Resulting external physical changes to the embodied system, (emission of light, movement of a limb, etc.), are collectively called the actuator output \((y \in Y)\) of this function. An actuator is any part of the system that can change the environment. A coupled pair of sensor input and actuator output may be described as a behaviour.

Let us define \(G\) as the intelligence function performed by the embodied system, mapping the input to the output. Where \(t\) is time, and \(\delta t > 0\) but arbitrarily small (in other words, time may be continuous—whether it is or not is a physical question that will be addressed in a future paper), we have:

\[
G_t(x_t) \to y_{t+\delta t}
\]

It is important to note here that, for our purposes, \(G_t\) can only be considered to cause an immediate actuator output (change that may effect the environment) as a result of an immediate actuator input (physical influence from the environment). It cannot be considered to implement any kind of plan over time, like commanding robot arm to move through a particular trajectory. Instead, the plan is carried out through \(G\) itself changing with time. \(G\) is altered by any internal changes to the system caused by the input (flow of a current inside a wire, charging of a capacitor, shifting of weight, etc.). The learning

http://www.aisb.org.uk
function, \( L_G \), determines how \( G \) changes with time: \( L_G(G_t, x_{t-\delta t}) \rightarrow G_{t+\delta t} \). This process is called adaptation.

This is a subtle, but important, difference in the way we think about how machines work. Were the intelligence able to issue a command to be followed by an actuator over time, some controller would have to be at work in the actuator to make sure that the command was carried out. This is fine, but in our system this controller is considered to be part of the intelligence function.

Now, let \( x' \) be the output from the environment to the system, and \( y' \) be the system’s input to the environment: the impact of the system’s behaviour on its surroundings. Where \( E \) is the environment reaction function mapping the input to the output:

\[
E_t(y'_t) \rightarrow x'_{t+\delta t}
\]

There is also an environment learning function, \( L_E \), equivalent to \( L_G \), that determines how \( E \) changes with time.

## 4 Categories of physical interaction

### 4.1 Real interaction

The interaction between \( E \) and \( G \) may be considered to fall into one of two classes. Real interaction is a pure physical process in which embodied intelligence co-evolves with its environment: such that the two functions are dependent only on initial conditions, their governing functions (\( L_E \) and \( L_G \)), their interactions with each other, and time. Thus, any adaptation of the embodied system is in direct response to its environment and nothing else (and vice versa). See Figure 1. This type of interaction requires that \( x = x' \) and \( y = y' \). Consequently, the domains/ranges of the two systems must be the same.

In the real-interaction scenario, as well as being a function, \( G \) may also be considered a description of the system’s instantaneous physical state at the arrival of the input stimulus. It is important to note that the state is here defined as not only specific parameters that can be measured instantaneously (speed, position, etc.), but also all associated rates of change. For example, two balls—one at rest and the other falling under gravity—would not be considered to have the same intelligence function even if they were identical in all other ways. Instantaneously they might look the same, but with associated rates of change taken into account, they are clearly not.

This should also highlight how intelligence in the physical sense is different from our idea of intelligence in computers normally. With conventional computers, we consider intelligence functions on subsets of stimuli and represented as abstractions. This makes them implementation independent: an adder can be built in many different ways with many different materials, but intelligence is required to determine what this intelligence function actually is and to interpret—which we can define as weeding out the relevant from the irrelevant—the results. With physical computation, the function and implementation are one and the same.

In the real-interaction case, implemented in the physical world, various constraints may be imposed. These include the following:

- The combined values of certain physical parameters of the two systems may be conserved (e.g., conservation of energy).
Figure 1: Here an object is evolving under physics. $G$ is the function that the object performs on the inputs ($x$) it gets from the outside world, which determine how it will impact the outside world ($y$). Likewise for $E$, which is the function performed by the outside world on the input from the object ($y'$), producing the output ($x'$). These two functions are carried out in parallel. Though labelled differently, $x$ and $x'$ must be the same for real interaction to take place, and likewise for $y$ and $y'$; thus the domain of each function must be the same as the range of the other.
Bains

- Changing functions and variables in the model may be constrained to evolve in certain ways: i.e. they may be constrained to vary continuously or to have particular allowed values.

- Since the only driving mechanism available is the function collectively known as \( \text{the laws of physics} \) (as they exist rather than as we understand them), \( L_p \), the constraint that \( L_E = L_G = L_P \) may be imposed.

In Section 5, we will consider some potential specific constraints of the laws of physics.

4.2 Virtual interaction

The second class of interaction to be considered is virtual interaction, which may be mediated by symbolic representation and communication (thus allowing the domains/ranges to be mismatched). Here, we define a symbol using Turing’s 1936 definition: a letter or sign taken from a finite alphabet to allow distinguishability. We define communication using Shannon’s communication theory: the sending of a message to a receiver via a (potentially) noisy channel (Shannon and Weaver, 1949). See Figure 2.

![Diagram](https://www.eish.org.uk)

Here, two processes are carried out in parallel: the virtual intelligence function, \( V' \), performs the intelligence task at hand, and the complementary function, \( V \), allows this arbitrary function to be performed whilst the entire system obeys the laws of physics. The latter plays many critical roles: it acquires information (sensing) and weeds out information from incoming signals that cannot be handled by the virtual machine (data reduction, domain conversion); it provides an interface to the real world (actuation) by mapping the machine state to an actuator state (control); finally, it allows communication between the various different parts of the machine.

Figure 2: In deterministic virtual interaction, real interaction is also taking place. However, the output from the environment is not the same as what is being input to the function that we deem intelligent: the virtual machine with function \( V \). For this to be the case, we model the interaction as including a complementary function \( (V') \), that filters the input from the outside world, feeds it into the virtual machine, processes the remainder, and combines the virtual machine result with its own to produce the final output. The virtual machine and its function are normally analyzed by computer scientists, whereas the complementary functions are the province of electrical, electronic, mechanical, and other engineers.

If we define \( V \) as the virtual intelligence function, analogous to \( G \) but for the virtual case, some major constraints are lifted, with important consequences. First, \( x \) is no longer
constrained to be equal to $x'$ (likewise for $y$ and $y'$). What this means is that the inputs and outputs may not be considered in their totality, but selectively (entire modalities may simply be excluded by $x$ and $y$, or specific ranges within specific modalities may be excluded). This can be considered in two ways. From a design perspective, it means that behaviours (sensor-actuator pairs) may be considered identical even if the global inputs and outputs (the inputs and outputs taking all possible modalities into account) are very different.

For example, when the letter A is typed into an laptop computer, how hard the key is pressed (within a range) is irrelevant. Soft A and hard A are considered, within the virtual environment, to be identical inputs. Likewise, the brightness of the screen is irrelevant to the meaning of the letter A when it appears before the user: whether or not our laptop is in power-saving mode does not affect our perception of the output here. So we have virtual sensor-actuator pairs (letter is typed, appears on screen) that can be very different physically but are considered to be the same behaviour virtually. In this scenario, there are fewer sensor modalities available to the embodied intelligence than there are actual modalities of physical influence coming from the environment.

As a result of this, the actual output from the environment may be very different from the input received by the virtual embodied intelligence. They may be very different because they are not allowed to affect the working of the virtual machine at all (just as how hard a key is pressed is information that is not available to the computing machine within the laptop). Or they may be represented very differently. For instance, temperature, which may be continuously varying in the environment, may be represented as a number with just one decimal place in the machine. Mathematically, in either of these cases, the domain and range of the two functions may be different.

In order for all of the above to be true, a new function must be defined: the complementary function $V'(x' - x)$ that ensures that, together with the intelligence function, the entire system obeys the laws of physics. The existence of this function can be considered to be a test of whether a system is virtual or not.

Another important constraint that is lifted is that $E$ and $V$ need not have any kind of conserved relationship, and $L_V$ need not be the same as $L_E$. Because only range/modality subsets of $x'$ and $y'$ attach to $V'$ and $L_V$, the intelligence and adaptation functions are partially de-coupled from the environment. With the right machine and choice of sensor modalities, arbitrary choice of $V$ and $L_V$ may be made.

It is important to note that this arbitrary choice depends on the selection of inputs/outputs, since a real interaction (with adaptation function $L_P$) must be taking place at the same time as this virtual interaction. Thus, it is only because of $V'$ and $L_V$ that such freedom is allowed for $V$ and $L_V$.

### 4.3 Non-deterministic interaction

The argument so far assumes (as is generally assumed in all branches of physics, except quantum mechanics, which will be discussed in more detail later) that the physical evolution taking place is both a causal and a deterministic process. In this context, causal means that the state of the two systems $E$ and $G$ at a given time $t$ is the direct and only cause of its state at time $t + \delta t$. In other words:

$$(E_t, G_t) \rightarrow (E_{t+\delta t}, G_{t+\delta t})$$

Deterministic means that the state at time $t + \delta t$ can be predicted from that at $t$. Non-deterministic means that it cannot. Note that there is an ambiguity inherent in this
Bains

definition. It is not stated by whom or by what this prediction can be made, nor in what circumstances. It may be unpredictable in principle because the process has some kind of random element, one not following any principle or order, embedded in it. Or it may be non-deterministic in practice: because insufficient information is available to make a reliable prediction.

It may seem that an interaction may not be both causal and non-deterministic. However, from the perspective of conventional quantum mechanics, the two are compatible in the following sense: physics (in the form of the wave equation) causes a particular set of outcomes to be possible, but which of these outcomes actually takes place is not determinable in advance. This is the definition of a stochastic process. The word random, left undefined in the last paragraph, can be more clearly understood in this context. If the process of “choosing” between different possible outcomes is not determined by physics, then it must be determined by something outside or above physics: meta-physics.

Two examples of in-practice non-determinism may make clearer the distinctions between this and the in-principle variety described above. First, a chaotic system might be practically non-deterministic to an observer if it were impossible to get infinitely precise information about it: however, there may be no principle that says that such information could not be made available. Second, a quantum-mechanical system might be considered unpredictable by scientists because the required properties and variables to make the prediction cannot be measured without changing them.

As the physical model presented in this thesis is not based on the manipulation of information or prediction based on a model, the in-practice scenario does not qualify as non-deterministic. Consequently, the model described in this section only applies if the in-principle variety of non-determinism is true. This distinction becomes important when the issue of how the mathematical model relates to real physics is considered.

4.3.1 Model of stochastic interactions

Mathematically, a causal, deterministic process can be represented by a one-to-one mapping from the input to the output: for a given set of conditions, only one outcome is possible. Both the real and virtual interactions described in the previous sections are based on this type of mapping. A stochastic or non-deterministic interaction, on the other hand, must be represented as a one-to-many mapping. In this case the output (x' or y) cannot be represented by a single value. Instead it must be taken from a set. In principle, this set may have any cardinality (with the possible exception of being empty): it may be finite, denumerable or non-denumerable. In Figure 3, for simplicity, the set of possibilities is kept to just two.

As can be seen in the figure, there is not (by definition) a pre-determined timeline. Specifically, knowing the state of one system at one time no longer uniquely implies the state of the other at that time, nor its own state any time later. This can be understood by following the various allowed evolutionary pathways for the two systems. As time elapses, the number of states that each system may have evolved into increases. This may be considered a type of divergence, in that it directly prevents tight coupling between the two systems.

To allow the mathematical model to take this type of evolution into account, a new random variable, z, can be defined that chooses which of the possible physical outcomes occurs: this variable is entirely independent of x, y, x', and y'.

Such a scenario may be considered to be a type of double virtual interaction. Functions G and E are constrained (by definition) to only act on input from each other (x, y, x', and y'). And yet, somehow, there is a function in existence that operates on z. As our model

http://www.aish.org.uk
Figure 3: In a non-deterministic universe, more than one outcome may be possible for a given physical event (the non-deterministic laws of physics are used here: \( L_{P}^{ND} \)). Two are shown here for simplicity. Thus, the coupling relationship between the two systems may change over time depending on which course (path) events take. Since the choosing of the path is metaphysical (not produced by object or environment), such an interaction cannot be considered real as it has been defined here.

is defined, the only way this is possible is if \( E \) and \( G \) are mapped to virtual functions \( V^{E} \) and \( V^{G} \) and their complementary functions, \( V^{IE} \) and \( V^{IG} \), are considered to keep track of \( z \) (see Figure 4). Here, \( V^{G}(x) \rightarrow Y \), and \( V^{IG}(y, z) \rightarrow y' \). Crucially, \( z \) is metaphysical here: its value is determined by some process that is not entirely dependent on, or related to, any of the physical laws or variables.

Given this analysis, only deterministic physical processes can be considered to fall into the class of real interactions as defined above.

http://www.aisb.org.uk
Figure 4: Instead, the non-deterministic case can be considered to be a kind of double virtual interaction, where the virtual machine is that obeying the deterministic physical laws that produce a set of possible outcomes, and the complementary function chooses from this set based on some metaphysical random variable, $z$. 
4.4 Real versus virtual interaction

To summarize this section, it is worth going back to our laptop example and comparing the real and virtual interaction analyses. In the former case, the machine may be considered just like any object. It is something that moves under gravity, has resistance, gives off heat, makes noise when touched etc. It can be used as a paper-weight or a doorstop. All of these uses of the machine are covered. In the virtual analysis, only the tapping of the keys and the pixels on the screen are considered to be part of the machine’s intelligence function. All other aspects of its behaviour, which combine to give it the global function $G$, are included in $V'$. Thus, for simulated robots, artificial life, and other applications where the interaction is purely symbolic and has no real meaning (apart from for display purposes) outside the machine, $V$ is all that’s important: the complementary function is irrelevant (as it is normally treated in computer science). For real robots in a real world, however, $G$ is critical. Thus, $V'$ must be taken into account when designing and building them.

5 The interface with physics

As described so far, the model is entirely general, in that we have not assigned the various functions or variables as belonging to any specific sets. It is only with this assignment that the representational and computational power of the model can be determined and (as required) compared to other computational models (such as the Turing machine). In order to provide the bridge between the theoretical and the physical, these assignments must come from our understanding of the physical laws. The relevant correspondences are outlined below.

The first important physical question that must be answered in order to allow comparison between various models has already been alluded to in Section 4: this is the issue of whether physics is entirely deterministic, or whether it is not. As discussed earlier, this information allows us to know whether real interaction actually exists, and therefore which model should be used.

Second, $x, x', y, y'$, are place holders for the multidimensional space that includes all the physical state variables, and $t$ is the placeholder for time. From a mathematical perspective we can ask two very simple questions about these variables that allow us to begin to understand their representational power. Are their sets finite or infinite? If infinite, do they have cardinality $\aleph_0$ or $\aleph_1$? These questions are crucial, because the answer determines whether or not the physical computation is less powerful, more powerful, or as powerful as (for instance) the Turing machine.

For instance, if it turns out that $G$ can map a continuous variable onto a continuous variable ($|x| = |y| = \mathbb{R}$) then this is the equivalent of a machine with an infinite symbol set and an infinite rule book: this is, specifically, not allowed with a Turing machine and represents the super-Turing case (it potentially contains all the mappings available to the Turing machine as well as others). For $|x| = |y| = \mathbb{R}$, and $|x| = |y| = N_{MAX}$ (a finite set), the machine would be Turing-equivalent, or sub-Turing, respectively.

With some rearrangement, the questions about cardinality can be turned into physical ones and then fed back into the model. First, are the multi-dimensional physical state variables and time—or, more simply, is space-time—continuous (i.e. there is a minimum $\delta x$ or $\delta t$), or arbitrarily discrete (i.e. there is no specific minimum $\delta x$ or $\delta t$ but the variable is still not defined as continuous). Second, are the physical systems in question bounded or unbounded? For this latter question, in the case of the embod-
ied intelligence, the answer is given in the definition: it is bounded. In the case of the environment, however, the question is still not decided among physicists.

We will leave this discussion here, as a full explanation of the relationships between the various physical and mathematical options is beyond the scope of this paper. However, we hope that it is, at least, clear that an interface between the theoretical and physical paradigms does exist here.

6 Discussion

In the brief discussion of comparisons above, a practical matter is not made explicit. The question we are asking is whether or not G may be replaced with a Turing machine. In fact, as discussed in Section 2, we know it cannot be, because there is no mechanism for getting information in and out of the machine from the real world. Thus, it must be the complementary function, V′, that allows us to feed our Turing machines (or digital computers) with the symbols they need, and to use the resulting symbols to produce an output. Thus, we can explicitly say that any symbolic interaction is, by definition, a virtual interaction, and the complementary function is crucial to its success. What, in practice, is V′′? It is the machine’s analogue sensors and actuators and enablers: mechanics, optics, hydraulics, heat-sinks, etc.

The same thing might also be said of our bodies, which not only provides our conscious mind with sensors and actuators, but also the un/subconscious mind which is not normally considered in AI. Other bodily systems—the limbic system, the spinal chord, even basic organs like the heart and lungs—all contribute to our behaviour to a greater or lesser extent. Using the virtual interaction model, we can see that what many people consider to be artificial intelligence, particularly purely software-based approaches, concerns only V and not V′.

What is interesting here is not only that this V′ exists, but that it may be as important as V for some applications.

This fact is increasingly being recognised by roboticists. For instance Williamson, who worked on Rodney Brooks “Cog” project, was charged with engineering the robot’s limbs in such a way as to ease both the information-processing and energy burden they represented for the machine as a whole. A mechanical engineer by training, he designed Cog’s arms and wrists so that they were compliant (Williamson, 1995) and could respond mechanically to changes in the environment rather than purely through conventional sensor-processor-actuator loops. The result was not only a more mechanically-efficient and natural-looking movement, but considerably lower computational overheads. Others interested in exploiting a balance between information processing and physical (in this case, mechanical) computation include Lungarella and Berthouze (2002), who have looked at how temporarily restricting the degrees of freedom of a mechanical system can improve a robot’s ability to learn how to manipulate it, and Pfeifer who gives numerous examples of the importance of mechanical design to machine intelligence (Pfeifer, 2002).

This balance may also be said to be the concern of the neuromorphic engineering field pioneered by Mead. This work often goes much further, blurring the interaction boundaries. Projects like Harrison and Koch’s all-analogue fly vision system and robot controller (Harrison and Koch, 2000), Hasslacher and Tilden’s analogue walking robots based on the nervous systems of small animals (Hasslacher and Tilden, 1995), or Lewis’s bipedal robots based on central pattern generators (Lewis et al., 2001), all have in common that there cannot be said to be a clear line between sensor, processor, and actuator. Indeed, there cannot be said to be a clear line between software and hardware. Whether their
systems can also be considered to perform real interaction or not is complex (and will not be considered here), but their machines certainly seem to address the complementary function more fully.

Finally, one could also argue that full understanding of real interaction, virtual interaction, and the complementary function might go some way to addressing Brooks’ recent question about the relationship between matter and life (Brooks, 2001). In essence, we may just need to go further to try to understand a whole brain and body (both V and V') in order to understand how a creature works, rather than considering mind alone.

7 Current and future work

From the description above, several projects suggest themselves: some of which are currently underway and others that might usefully be done in future. The first of these, currently in progress, is the analysis of the model under constraints as suggested by current (and often contradictory) interpretations of the physical laws, and comparisons with the Turing model given the options that arise from these. In particular, issues related to quantum mechanics are problematic. Whether the universe may be considered discrete or continuous is not a trivial problem: there is no consensus here. Likewise, there are still many who believe that the current consensus in quantum-mechanics—of which it may be said that the theory is sound, but the philosophical underpinnings not—will eventually have to change. We are currently developing a map of these ideas so that, as the physics develops, its impact on the question of intelligence can be seen clearly. This work is currently being written up.

We are also trying to identify those applications of intelligence where a mismatch in representational power may be important. Clearly, information-theoretic approaches are appropriate in many engineering scenarios: today’s engineering is based on such approaches. We are currently drawing up a simple specification, in terms of the types of functions that may be important and the type of stimuli that may need to be represented, that should allow engineers to understand where a more physical approach may be warranted. This work is also at the draft stage.

As a longer term goal, we would like to be able to help the engineer who, using the test outlined about, has determined that the conventional design approach is inappropriate for the task in hand. This involves the development of a set of design rules that would determine how analogue to digital conversion layers are placed in a given system: i.e. how to correctly balance the analogue and digital processes in order to maximise efficiency for a given task. Note: in some cases it might be expected that no conversion is the best option, thus suggesting an analogue-only system.

Acknowledgements

The author would like to acknowledge: her supervisor at the Open University, Jeffrey Johnson; the help of the reviewers; Christof Koch of the California Institute of Technology for important discussions and reading the manuscript; her colleagues at Imperial College; and numerous other scientists and engineers who have contributed to this work through inspiration, discussion, and debate.

http://www.aisb.org.uk
Bains

References


Physical Computation


http://www.aisb.org.uk
Appendix 5:

Sunny Bains and Jeffrey Johnson

*Noise, physics, and non-Turing computation*

Joint Conference on Information Systems

Noise, physics, and non-Turing computation

Sunny Bains† and Jeffrey Johnson
Department of Design and Innovation
The Open University
Milton Keynes, MK7 6AA, United Kingdom
†Corresponding author.
E-mail: sunny@sunnybains.com

Abstract
It has been proven that recurrent neural networks with analog weights can solve problems that are insoluble using Turing machines. However, the prevailing view of those involved in computing research is that a) non-Turing computation is unnecessary for intelligence and b) machines with non-Turing abilities cannot be built in practice. Two reasons are commonly cited for this. First, the effects of noise are said to remove any possibility of using an analog system to represent quantities with continuous infinite precision. Second, the (asserted) discrete nature of the Universe, is cited as proof that truly analog systems cannot exist. These are issues of physics that exist well outside the domain of computer science and engineering. Here, we will examine the technical and mathematical bases of these claims, suggest counter arguments, and discuss the deep physical questions that they raise.

Introduction
Siegelmann has shown that recurrent analog neural networks (RANN) [Siegelmann] have, at least theoretically, the ability to perform computational tasks that go beyond those of the Turing machine. This could help to justify an analog version of computationalism: the position that human-like intelligence is possible in computers. In particular, by removing computers from the confines of the Turing machine [Church] [Turing], many of Penrose's well-known objections to AI [Penrose] would become moot. In this sense, it is not a counter argument (of which there are many in the literature, see, for example [LaForte]), but rather a solution to the problem Penrose raises. ([Deutsch] suggests a partial solution based on quantum computing, but this construct—the universal quantum computer—is still not comprehensive in its power: it cannot compute non-recursive functions.)

The practicality of the super-Turing finding has been questioned. Some researchers claim that analog (as opposed to digital) computation is a meaningless concept because digital computers can always attain a sufficient level of precision to prevent there being any difference in outcome. Others claim that noise prevents real neural networks from reaching their alleged potential (see, for example, [Maass]). Though there are many such analyses, most have in common that they come directly from interpretations of Turing universality and communications theory.

This paper will discuss the validity of using such theories to describe continuously-evolving physical systems—whether electronic or biological—in which the processes taking place cannot be described in terms of conventional measurements. First, we will cover some basic arguments in favor of super- or non-Turing computation. Next, we will look at the objections to computation beyond the Turing limit. Then, we will examine the underlying physical assumptions that lead to these objections, after which we will describe what we believe to be a more physically valid interpretation of the way that recurrent analog neural networks interact with the outside world. Finally, we will explain how, as Penrose has always contended, the understanding and interpretation of quantum mechanics lies at the heart of the problem of computational intelligence: at least from a theoretical point of view.

The mathematical basis for super- and non-Turing computation
From a mathematical perspective, allowing neural networks to have weights that vary continuously clearly allows for a richer representational ability than a machine that uses a finite number of discrete symbols on an infinite piece of tape (part of the definition of a Turing machine). Though we will not go into the details here, the following mathematical arguments are straightforward to prove.

If a Turing machine is to process in a finite time, and each square or digit on the tape takes a finite time to read, then only a finite portion of the infinite tape can actually be used. This limits the device to dealing with a subset of rational numbers (those rational numbers that can be represented by the particular number of symbols or bits that can be read in the allotted time). To represent a continuous-valued quantity, the set of real numbers, $\mathbb{R}$, must be mapped on to (at most) the set of rational numbers $\mathbb{Q}$. There are more irrational (of the set $\mathbb{Q}'$) than rational numbers. Though both sets are infinite, the former has cardinality $\mathfrak{c}$, whereas the latter has cardinality $\aleph_0$. As a result, many states in the

* There is an argument that you can avoid this problem by using geometrically decreasing times, eg. processing the first digit in time $t$, the second in $t/2$, the third in $t/4$, etc. This produces a series that converges to $2t$. However, this has a problem of specificity: in this case opposite to that of the arbitrary precision argument discussed on the next page. In the former, the ability to process an infinite number of squares disappears as soon as one specifies a particular minimum processing time per square. In the latter the arbitrary precision is only sufficient as long as the data is specified in advance.
original continuous function map to a single number within the Turing machine. This loss of information between the continuous-valued world and the Turing machine is the source of the proposed non-Turing abilities of the RANN (for a full proof and a discussion of what these abilities are, see [Siegelmann]).

**Objections**

For any given process, the Turing machine can be designed use enough bits to produce the correct answer. This is only true if the designer knows in advance what the required precision is. For nonlinear functions, the data itself determines how much precision is required. Take the sigmoid function, for instance, going from $-1$ through $0$ to $+1$. If the magnitude of the incoming continuous signal is relatively large (far from the threshold level $0$, then it can be represented very inaccurately without changing the output. For inputs of smaller magnitude, however, more precision is required. The difference between $+0.0001$ and $-0.0001$ in the input, may make a huge difference to the output. This problem increases as inputs get closer to zero. To accurately represent the incoming signal therefore requires foreknowledge of its value. While this is feasible for many engineering applications, it is not practical for intelligent systems designed to function in an unforeseeable universe.

Quantum mechanics says that the universe is discrete, not continuous, as evidenced by the existence of the Planck length and time. Thus, there is a finite level of precision beyond which it is meaningless to go. Even if length were discrete, position could still be a continuous variable. Imagine a watch with a second hand (of $n$ Planck lengths, where $n$ is an integer) tracing out a circle. Consider that a particle at the end of the second hand moves discretely: one Planck length in circumference of a circle is $\pi r$ times the Planck length. Because $\pi$ is an irrational number, not only there will there be an offset from the starting position in the first cycle to that of the second, but there will be an offset for every cycle. Further, the particle cannot go through the same position twice because it can only move an integral number of Planck lengths in distance. For it to take the same position on the $x$th turn as it did on the first, therefore, $n \cdot 2\pi n$ has to be an integer. An irrational number multiplied by an integer is always irrational, so this is impossible. Thus, if the second hand is allowed to turn an infinite number of times it must pass through an infinite number of positions. Position must, therefore, be considered a continuous variable.

Regardless of the precision that a Turing machine can or cannot achieve, a practical analog machine can do no better because of noise. Where a digital computer loses information by mapping a continuous variable onto a discrete set of numbers, an analog computer loses information by mapping any given input point onto many output points (a different output for each noise value). The super- or non-Turing machine is therefore physically unrealizable. See, for example, [Hadley]. Though recent results [Mar] have shown that, in fact, coupled neural networks are much less sensitive to noise than had previously been thought, this objection gets to the heart of the controversy concerning super-Turing computation. One weakness of this position that no-one has, so far, proven that noise removes all non-Turing abilities from the RANN (though it has been proven that some of the Turing abilities are lost, eg. [Maass]). It is possible that we are not forced to choose simply between Turing and super-Turing abilities, but that there is a realizable subset of the latter that can perform some Turing and some non-Turing functions. However, this question remains unanswered, so it does not really help us here.

The analysis of noise as a degrading element in recurrent analog neural networks is a natural result of communications theory [Shannon]. Communications theory is an extremely important foundation for many fields in computer science and electrical and electronic engineering. It is an important tool, but relies on the following assumptions (see Figure 1A):

![Figure 1](image)

1. There is a *sender* who is trying to convey a particular piece of information or *signal* down a *channel* containing *noise* to a *receiver*.
2. The noise is unrelated to the sender or receiver.

Using communications theory to analyze a digital computer or analog communications system is...
appropriate. However, in the next section we will discuss why a recurrent analog neural network can be considered to be a qualitatively different kind of system or process: one that must be analyzed in a different way.

Physical computation
Consider a brain, whether electronic or biological, as having the following properties: it obeys the laws of physics and it is structured in such a way that it is particularly sensitive to, and able to adapt to, the rest of the universe. Consider that, at time \( t \) the brain is in state \( B \). By this I mean that, were it possible to measure all the physical variables relating to all the particles that comprise electrons, atoms, molecules, neurons etc., respectively, they would be described by \( B \). (The fact that this quantity is, in principle, unmeasurable, is irrelevant to our argument). Consider that, at time \( t \), the rest of the universe could be described by a similar state variable \( U \). These two "entities"—brain and universe—exert forces on each other and exchange particles and energy as dictated by the laws of physics (see Figure 1B).

Let us now assume that the universe is deterministic and non-random (we will come back to this assumption later). In this case, as \( B \) and \( U \) evolve over time, they will be intrinsically coupled. By this I mean that if "outside observers" had knowledge of \( B \) at time \( t_1 \), and complete understanding of the laws of physics, then by looking at \( B \) again at time \( t_2 \), they could infer something about \( U \) without observing anything but the brain. This information would be incomplete or ambiguous, in that many different states of the universe could cause the same change in the way the brain evolves with time. However, there is no sender, no message, no receiver, no channel, and therefore no channel noise here: simply two co-evolving physical systems.

The aforementioned ambiguity is the only source of information loss in this situation. Noise is not an issue at this point because it is a property that can only emerge through a statistical model of some phenomenon (such as thermodynamics). In such models, we make an engineering trade-off: we free ourselves of the need to know the microscopic details of a situation (like the position and momentum of individual gas molecules) in exchange for some uncertainty in our ability to predict. There are no such compromises made in our scenario. All variables are specified, all forces are known. All that the universe and brain have to do is to obey the laws of physics and they will evolve from their initial conditions with infinite precision.

At this point we should point out that the situation outlined here is as true for an inanimate object as it is for the brain. The object, left to its own devices, will evolve exactly as the laws of physics dictate. There is no opportunity for information to be lost.

Next, consider what makes the biological brain, for instance, different from an inanimate object: the fact that the former is physically structured to evolve in such a way that its ability to survive increases over time, allowing it (and the animal that's attached to it) the opportunity to procreate. (This is not to suggest that intelligence is a passive process, merely that the activity is determined, like everything else, by physical laws). Let us consider, that the brain is structured as a recurrent analog neural network (RANN) and includes sensory organs (ie. that the brain is embodied). First, let us note that just because the system is designed to be particularly adaptive to particular sorts of stimuli (eg. light, sound), that does not mean that it is only sensitive to these phenomena. Gravity, invisible radiation (to the eye), magnetic fields etc. all have some influence on the state of the matter that makes up the brain: even if that influence is small (affecting microscopic rather than macroscopic events). Though the effect may be subtle, these events will affect the operation of the neural network. Because learning is going on in the presence of this underlying microscopic activity, that must somehow be reflected in what is learned. Thus these phenomena represent a source of information about the universe just as much as incoming light rays do.

Second, the brain has no way of distinguishing between "signals" and anything else, except by experience. The brain's job is merely to map the right set of responses to the external conditions, however those happen to manifest themselves. Information about temperature, for instance, encoded as what engineers might call "thermal noise", may be just as important for determining a particular outcome as whether the sky is light or dark. Thus, "thermal noise" is a signal for that task. (Likewise, there may be cases where, for instance, aural inputs are irrelevant to the task at hand and the brain can learn from experience to ignore them for that purpose). Thus what constitutes noise is purely context dependent in the physical brain.

In other words, it is only when we attempt to model these brain functions that we are forced to describe as noise those things that we cannot model: either because they are unmeasurable or because they complicate the calculation. Looking from the opposite point of view, consider the engineer trying to model a particular neural network. One can model the function of an individual neuron, the constituent molecules, the electrochemical components, down to the quantum-mechanical level. At each stage uncertainty (or noise) is stripped away, the trade-offs being increased complexity in the model and increased precision required of the initial conditions. At the lowest level, that of quantum mechanics (and related theories), the noise disappears entirely (as everything is explicitly known) except for quantum noise.

For the engineer, this precision is of course impossible. According to the uncertainty principle,
there is no way to make the measurements required to predict the behavior of the neural network. However, the engineers only job is to make sure that the neural network (or brain) is sensitive to the universe at large and able to adapt in useful ways. After that, physics takes over. There are no explicit measurements, no calculations: the analog network can simply evolve according to the physical laws (which it must, and will, do precisely). It is this physical computation that is not subject to conventional types of noise.

The statistical nature of Quantum Mechanics

If the super- or non-Turing properties of recurrent analog neural networks are still to be eroded by the presence of noise, that noise must be quantum mechanical. Further, that quantum mechanical noise may not simply be caused by observational unpredictability (where the observer simply lacks enough information determine what will happen next) or problems of measurement (where the measurement itself changes the system, thus preventing the original quantity of interest being accurately determined). In order for the quantum noise to be truly noisy, it must be unpredictable by physics itself.

There are two interpretations of quantum mechanics that allow this true unpredictability. First, the so-called "many worlds" theory suggests that all statistical possibilities for any particular quantum event do in fact happen. Our universe simply takes a "random walk" through these events. Another possibility is that there is some intrinsic source of randomness in the universe: so that some kind of "coin toss" takes place to determine every quantum event. These true noise sources would act to "decouple" the evolving systems, both brain and universe, degrading the information that each had obtained from its physical interaction with the other.

However, there are also theories (see, for example, [Bohm]) that allow for quantum noise to be unpredictable to observers but not to physics. The uncertainty principle prevents us from making sufficient measurements to make good predictions but, according to these theories, the underlying physical properties are still there and determining the outcome. No "coin" must be flipped. physics merely evolves according to quantities we cannot measure and laws we don't yet understand. In this situation, there is no such thing as noise.

Discussion and further work

The physics community currently favors the "random" interpretation of quantum mechanics, which could indeed prevent the existence of super-Turing computation. However, quantum mechanics is still not well-understood as a physical phenomenon (as opposed to a mathematical framework) so it is premature to use this result to answer the computational question. In addition, there seem to be physical phenomena that perform functions (or computations) that are computable neither by Turing machines nor by universal quantum computers. Quantum computation over continuous variables has been proposed, but because these systems still involve programming (setting initial states) and then explicit measurements, they have been found to be equivalent to the universal quantum computer [Lloyd]. Further examination of the non-simulable should shed light on the prospects for physical (non-algorithmic) computation.

The authors would like to thank the many scientists and engineers who generously contributed to this work through reference, discussion, and criticism.

References


Church, A (1936), An unsolvable problem of elementary number theory, American Journal of Mathematics 58, pp. 345-363.


Lloyd, S and S L Braunstein (1999), Quantum computation over continuous variables, Phys. Rev. Lett. 82 (8), 22 February.

Maass, W (1997), Analog Neural Nets with Gaussian or other Common Noise Distributions cannot Recognize Arbitrary Regular Languages, Electronic Colloquium on Computational Complexity Report TR97-052.


Penrose, R (1989), The Emperor's New Mind, Oxford University Press.

Shannon, C E (1949), W Weaver, The mathematical theory of communication, University of Illinois Press.

Siegelmann, H T (1999), Neural Networks and Analog Computation: Beyond the Turing Limit, Birkhäuser Boston.