Evolution in Materio: Investigating the Stability of Robot Controllers Evolved in Liquid Crystal

Simon Harding and Julian F. Miller

Department of Electronics, University of York, York, UK
{slh, jfm}@evolutioninmaterio.com
http://www.evolutioninmaterio.com

Abstract. In our previous work, we have demonstrated that evolution can be used to program liquid crystal to act as a signal processing device. In this work we discuss the stability and reconfigurability of a real time robot controller evolved in liquid crystal. We envisage these issues will be important when programming or evolving in other physical systems.

1 Introduction

Allowing computer controlled evolution (CCE) to manipulate novel physical media can allow much greater scope for the discovery of unconventional solutions. Last year the authors demonstrated, for the first time, that CCE could manipulate liquid crystal to perform signal processing tasks (i.e frequency discrimination, robot control). In [4] Harding and Miller showed that liquid crystal could be used as a medium for evolution. They were able to rapidly evolve simple transistor like behaviour and in [3] they demonstrated that it was relatively easy to evolve a liquid crystal to discriminate between pairs of dissimilar frequencies. The task was first considered by Adrian Thompson (using an FPGA) [10]. Recently we have investigated other tasks including robot control.

Here we examine some practical issues relating to evolving devices in liquid crystal. In particular we look at these issues in the case of evolving a real time robot controller for obstacle avoidance. We found that solutions were relatively unstable, and were greatly influenced by previous configurations. In this paper we investigate this phenomenon in detail.

1.1 The Field Programmable Matter Array

In [8] a device that the authors referred to as a Field Programmable Matter Array(FPMA) was described. A FPMA is a device that can be used to manipulate a material under computer control by applying voltages that induce physical changes within a substance, and that these changes may interact in unexpected ways that may be exploitable under evolution.

Different candidate materials were cited for possible use as the evolvable substrate in the FPMA. They all share several characteristics : the material should be configurable by an applied voltage/current, the material should affect an incident signal (e.g. optical and electronic) and should be able to be reset back to

J.M. Moreno, J. Madrenas, and J. Cosp (Eds.): ICES 2005, LNCS 3637, pp. 155–164, 2005.

its original state. Examples of these include electroactive polymers, voltage controlled colloids, bacterial consortia, liquid crystal, and nanoparticle suspensions. In previous work we have demonstrated that liquid crystal is indeed a suitable material to form the basis of the FPMA.

Liquid crystal (LC) is commonly defined as a substance that can exist in a mesomorphic state [1,7]. Mesomorphic states have a degree of molecular order that lies between that of a solid crystal (long-range positional and orientational) and a liquid, gas or amorphous solid (no long-range order). In LC there is long-range orientational order but no long-range positional order.

2 The Liquid Crystal Evolvable Motherboard

The Liquid Crystal Evolvable Motherboard (LCEM) is circuit that uses four cross-switch matrix devices to dynamically configure circuits that connect to the liquid crystal. The switches are used to wire the 64 connections on the LCD to one of 8 external connections. The external connections are: input voltages, grounding, signals and connections to measurement devices. Each of the external connectors can be wired to any of the connections to the LCD.

The external connections of the LCEM are connected a PC's analogue inputs and outputs. Connections can be assigned for the input signals, measurement, and for fixed voltages (plus a ground connection). The value of the fixed voltages is determined by a genetic algorithm[6], but is constant throughout each evaluation.

In the experiments presented here, the liquid crystal glass sandwich was removed from the display controller it was originally mounted on, and placed on the LCEM. The display is a passive monochromatic matrix LCD with a resolution for 180 by 120 pixels. Unfortunately neither the internal structure nor the electrical characteristics of the LCD are known. The display has a large number of connections (in excess of 200), and is roughly positioned over the pads on the PCB, with many of the PCB pads touching more than 1 of the connectors

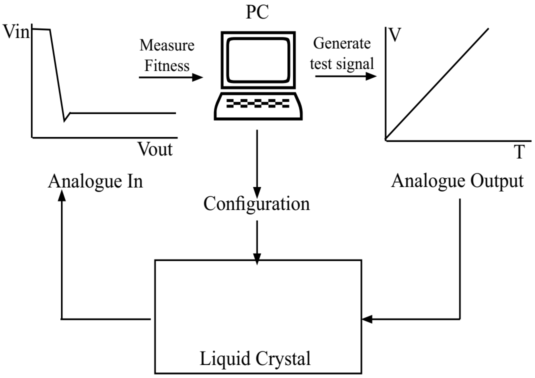

Fig. 1. Equipment configuration

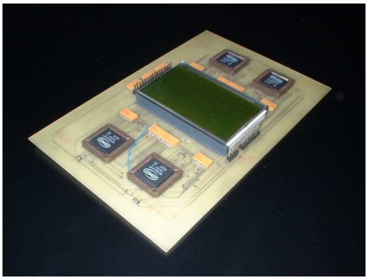

Fig. 2. The LCEM

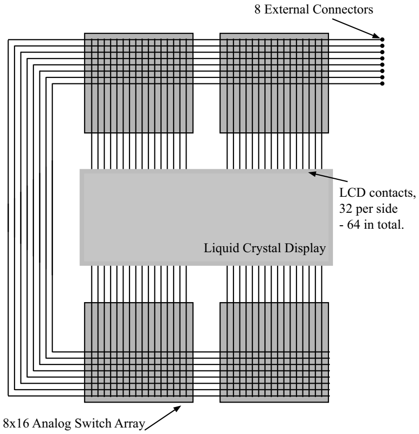

Fig. 3. Schematic of LCEM

on the LCD. This means that we are applying configuration voltages to several areas of LC at the same time.

It is important to note that other than the control circuitry for the switch arrays there are no other active components on the motherboard - only analogue switches, smoothing capacitors, resistors and the LCD are present.

3 A Liquid Crystal Robot Controller

In these experiments we used a simulated robot that has two sensors (mounted with 30 degrees of separation) and two wheels for mobility. The simulated sensor readings are converted into signals fed to the evolvable motherboard. Signals read from the evolvable motherboard are then used to control the behaviour of the simulated robot. The intention being that the signal processing, and majority of the robot control should be performed in the liquid crystal. Two sonar distances sensors and two motors can be considered to be "directly" connected to the evolvable motherboard, and then routed to the liquid crystal.

We defined each distance sensor to output a square wave with a frequency proportional to the distance in a straight line from the sensor to an obstacle. For near objects the output was 1Hz, for far objects the output frequency is 5000Hz. No artificial noise is added to the distance measured, however the mechanism by which the waves are generated by the computer will add noise and timing problems. There is also an expected 50ms delay between a distance reading and a change in frequency.

Two connections from the LCEM are used as inputs to the two motor controllers. The two motors are mounted either side of the simulated robot, and allow for the robot to be steered. If the voltage is high (i.e. above 0.3V) a motor is switched fully on, when low the motor is set to a slow speed. If both inputs are high the robot drives forward, with both inputs off the robot is stationery. If only one motor is enabled, the robot turns. The threshold voltage for enabling a motor was chosen arbitrarily. The robot has a small turning circle, and does not pivot on the switched off wheel.

3.1 The Genetic Algorithm

The genetic representation for each individual is made of two parts. The first part specifies the connectivity; the second part determines the configuration voltages applied to the the LCD. Each of the 64 connectors on the LCD can be connected to one of the eight external connectors or left to float (see Figure 3). Each of the connectors is represented by a number from 0 to 7 and no connection is represented by 8. Hence the genotype for connectivity is a string of 64 integers in the range 0 to 8. The remainder of the genotype specifies the voltages applied to the pins on the external connector that are not used for signal injection / monitoring. One of the external connectors is always connected to ground. Two are reserved for the incident signals (distance readings) and two connections for motor control. The remaining three connectors have static voltages applied to them that are determined by evolution. All these connectors can be routed to various places in the liquid crystal display according to the connection scheme decided by evolution. Each voltage is represented as a 16-bit integer, the 65536 possible values map to the voltage levels output from -10V to +10V. The second section of the genotype is therefore represented as a string of three 16bit integers.

In all the following experiments, a population of 40 individuals was used. The mutation rate was set to 5 mutations per individual. Elitism was used, with 5 individuals selected from the population going through to the next generation. Selection was performed using tournament selection based on a sample of 5 individuals. Evolutionary runs were limited to 200 generations. With each individual taking approximately 60 seconds to evaluate. The fitness function rewarded controllers that were able to travel around the environment without colliding with obstacles and for exploring as much of the environment as possible.

Further details of the fitness function, genotype and related operators can be found in [5].

3.2 Results

The fitness function rewarded perfect solutions with a score of 10000, with 0 as the lowest possible score. Solutions that have a fitness of over 6700 represent robots that have navigated to leave the top section of the map. Solutions below this score fail to fully explore the map - however they may cover large areas of the top half of the map but never escape through the gap. In our evolutionary runs we found 36% of runs obtained a near perfect score. The average number of generations to find a good solution is 62, with the fastest solution found within 22 generations.

Figures 4 to 9 shows sections of the "fossil record" of the evolution of one controller. We can see that after learning not to drive in circles, the robot learns to move forward, and then learns to turn when it approaches a wall. After it

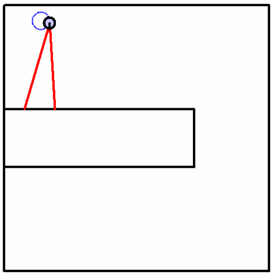

Fig. 4. Fitness=515

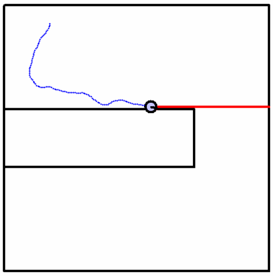

Fig. 5. fitness=3819

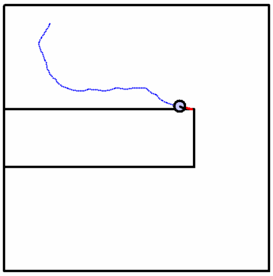

Fig. 6. fitness=4607

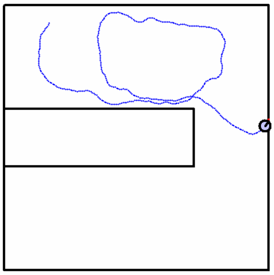

Fig. 7. fitness=6772

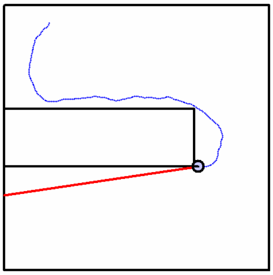

Fig. 8. fitness=7229

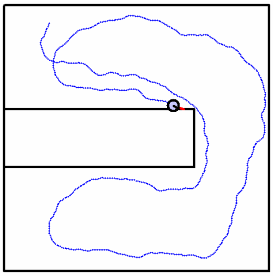

Fig. 9. fitness=9796

learns to start following the wall it quickly searches the entire map, and gets the highest fitness.

4 Investigating Solution Stability

When incident signals are applied to the liquid crystal display, we can see their effect - some of the pixels go dark - indicating that the molecular direction has been changed. This means that the configuration of the liquid crystal is changing as we are applying signals. To draw an analogy with circuit design, the incident signals would be changing component values or changing the circuit topology, which would have an effect on the behaviour of the system. This is likely to be detrimental to the measured performance of the circuit, and also we expect to a liquid crystal solution. When a solution is evolved the fitness function automatically measures it stability over the period of the evaluation. Changes made by the incident signals can be considered part of the genotype to phenotype mapping. Solutions that cannot cope with their initial configurations being altered will achieve a low score. However, the fitness function cannot measure the behaviour beyond the end of the evaluation time - however a stable solution is still desirable.

Another issue of stability is that of the genotype to phenotype mapping. When a configuration is applied to the liquid crystal, do the molecules go back to exactly how they were when this configuration was tried previously? We cannot be sure - since we cannot directly measure the properties of every molecule, and in a highly complex system such as LC it would be unlikely to reorder in precisely the same way. Assuming, that there is a strong correlation between genotype and phenotype, then it is likely that evolution will cope with this extra noise. The fact that it is possible to evolve in liquid crystal, shows that we should expect good genotype/phenotype correlation, however as the results in this paper indicate, this is not the case.

In [9] it is noted that the behaviour of circuits evolved intrinsically can be influenced by previous configurations - therefore their behaviour (and hence fitness) is dependent not only on the currently evaluated individuals configuration but on those that came before. For example, in a circuit capacitors may still hold charge from a previously tested circuit. This charge would then effect the circuits operation, however if the circuit was tested again with no stored charge a different behaviour would be expected and a different fitness score would be obtained - and the fitness function would essentially be noisy. Not only does this effect the ability to evolve circuits, but would mean that some circuits are not valid. Without the influence of the previously evaluated circuits the current solution may not function as expected. The behaviour does not have to be worse when dependent on previous configurations - there is no reason why the previous configuration cannot have a positive influence. It is expected that such problems will have analogies in evolution in materio. The configurations are likely to be highly sensitive to initial conditions (i.e. conditions introduced by previous configurations), as the ability to configure a system is reliant on the emergent properties of the material. The behaviour of emergent systems, such as Conway's "Life" [2] or Wolfram's cellular automata [11] are highly dependent on the starting configuration. Small perturbations in the initial starting arrangement can prevent solutions from becoming stable.

In these experiments, we investigate the stability of the solution, by looking at the performance of the solution for extended periods of time and the effect of previous configurations on the behaviour of the system.

4.1 Observing Continued Behaviour

The wall avoiding robot task often produced solutions that would take approximately 30 seconds to evaluate before the robot collided with an obstacle. This raised the issue of for how long are solutions stable? This experiment investigates the performance of solutions over a period of time. If the liquid crystal was being affected by the incident signals, and being reprogrammed, then it would be expected that the behaviour would change. A change in behaviour would change the fitness score measured by the fitness function. The fitness function used for this task returns an absolute fitness value - any change in behaviour should result in a different fitness value.

In these experiments robot controllers were evolved (as described in section 3) and when an individual received a high fitness score we repeatedly ran the evaluation function and recorded the fitness for the subsequent evaluations. The

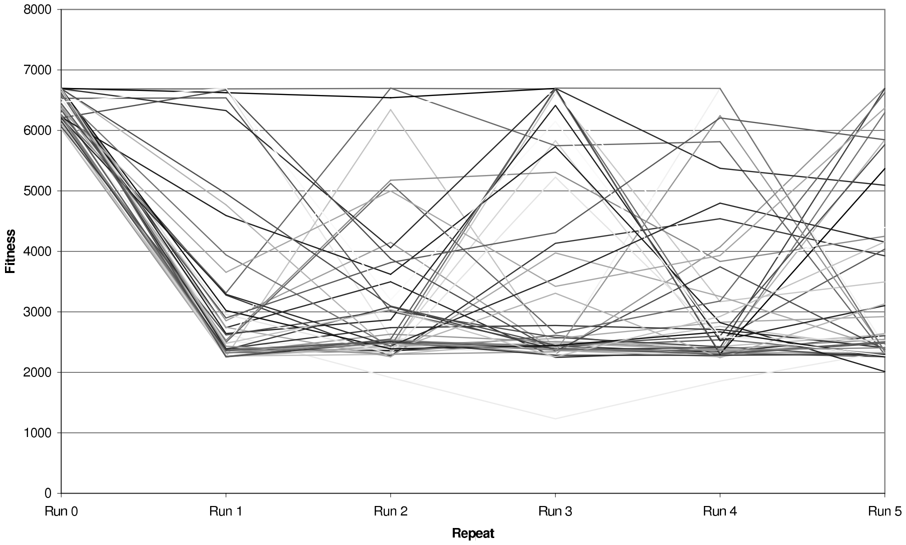

Fig. 10. Graph showing degradation of fitness when individuals are reloaded and tested, as described in section 4.1

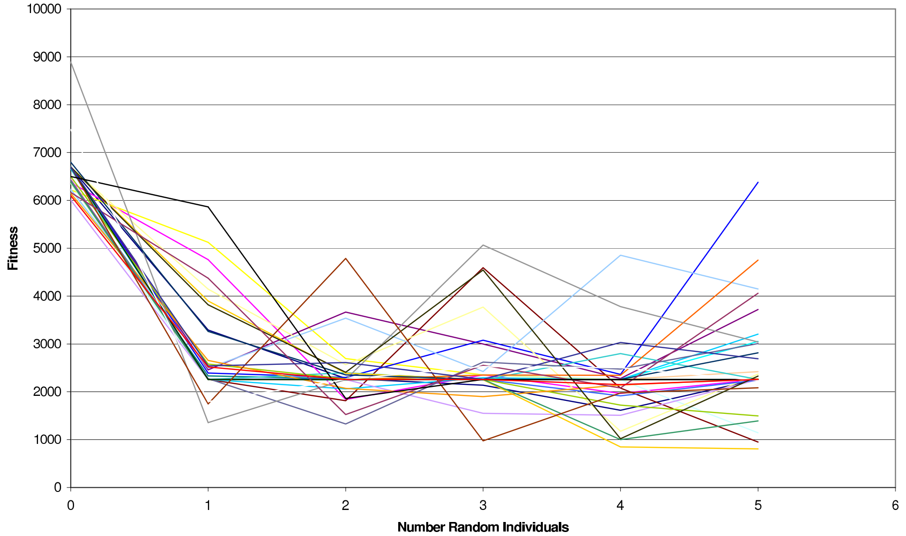

Fig. 11. Graph showing degradation of fitness when individuals are reloaded and tested, with intermediate individuals loaded in between, as described in section 4.2. The number of intermediate random individuals increases on each reload.

individuals were re-evaluated 10 times - giving a total running time of approximately 5 minutes.

4.2 Drift and the Reloading of Configuration

This experiment demonstrates the affect that previous configurations have on the behaviour of the current evaluation. The experiment is similar to that in section 4.1, however the configuration of the liquid crystal is modified in between evaluations. This was done by applying a number of random configurations, and then reapplying the configuration specified in the current individual. The intent is to disrupt the liquid crystal as much as possible, and to try and randomize the molecular configuration. This evaluation process can be summarised as:

```
1. Apply individual, and test fitness.
2. If the solution is good (i.e. with fitness>6700) then continue
   otherwise, goto step 1 and evaluate next individual.
3. N = 0
4. Apply N random Configurations to LC
5. Apply individual
6. Test and record fitness
7. N = N + 1
8. If N<6 repeat from step 4.
```

If configurations are dependent on previous configurations then the fitness values would be expected to be affected, and as there are few solutions the fitness would be expected to decrease. If the configurations are independent of each other, then the fitness results should be relatively consistent throughout the evaluation procedure.

4.3 Results

From Figure 10, it can be seen from the reduced fitness scores that most solutions fail to operate over long periods of time. However, some solutions do continue to function correctly for several evaluations. The results also show that the behaviour can deteriorate, and then recover. From Figure 11 it is apparent that previous configurations have a large effect on the behaviour of the system. Applying other configurations causes the evolved behaviour to worsen in every case. The fitness does vary through each iteration, however it never returns to the level initially achieved.

5 Conclusions

The effects observed in these experiments may preclude this set-up from having any real practical application. The issue of reliable genotype to phenotype mapping and reliable behaviour from the phenotype are serious problems. However, these are preliminary results from our first attempt at direct evolution in material. With further investigation, we believe that there will be methods to overcome these problems.

The most obvious solution is to find a more stable material. Liquid crystal was chosen as a candidate material because of the ease at which we could manipulate it. However there may be more suitable materials, that can be made less sensitive to disruption caused by incident signals.

The next potential solution is to evolve stable solutions. In the experiments in this section, solutions were stable only for a few seconds - however there was no pressure imposed for evolution to find solutions that had any robustness. With the robot controller, solutions had to remain stable for a much longer period - typically 30 seconds. It is apparent when forced to produce time-robust solutions evolution in liquid crystal is capable of doing so.

Another approach, for certain tasks, may be to find a different way to interact with the material. At the moment the method of communication with the liquid crystal is through different frequencies of electrical signals. This will have the effect of continually reconfiguring the system. It is not clear if this is the most appropriate way to present information to the system.

Liquid crystals are normally associated with displays as they can modify light passing through them. We can assume that light does not effect the liquid crystal and is only affected by it. If we used a camera to observe changes in light coming through the display we could see any differences in the properties of the light and use this as an output. Liquid crystal is also able to change the properties of sound waves passing through it. This technique is normally used to study the structure of the liquid. If the structure is modified using computer controlled evolution, it should be possible to alter the effect it has on sound waves. Incident signals could be applied using a speaker, and the output detected with a microphone.

Despite the issues of stability, the system is still evolvable. It would appear from the experiment in section 4.2 that there is little inheritance of behaviour from one generation to the next - as applying similar (and even the same) configurations may not result in the same phenotypic behaviour. However, the evolutionary algorithm still succeeds in finding solutions despite the highly noisy search space. In future we intend to investigate this further, and determine if there are any properties of this system that could be utilised by other evolvable systems.

References

1. D. Demus, J Goodby, G W Gray, H W Spiess, and V Vill, editors. *Handbook of Liquid Crystals*, volume 1,2A,2B,3. July 1998.
2. Martin Gardner. Mathematical games: The fantastic combinations of john conways new solitaire game life. In *Scientific American*, volume 223, pages 120–123, 1970.
3. Simon Harding and Julian F. Miller. Evolution in materio: A tone discriminator in liquid crystal. In *In Proceedings of the Congress on Evolutionary Computation 2004 (CEC'2004)*, volume 2, pages 1800–1807, 2004.
4. Simon Harding and Julian F. Miller. Evolution in materio: Initial experiments with liquid crystal. In *Proceedings of 2004 NASA/DoD Conference on Evolvable Hardware (EH'04)*, pages 298–305, 2004.

5. Simon Harding and Julian F. Miller. Evolution in materio : A real time robot controller in liquid crystal. In *To appear in the proceedings of 2005 NASA/DoD Conference On Evolvable Hardware*, 2005.
6. John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* MIT Press, Cambridge, MA, USA, 1992.
7. I. C. Khoo. *Liquid Crystals: physical properties and nonlinear optical phenomena.* Wiley, 1995.
8. J. F. Miller and K. Downing. Evolution in materio: Looking beyond the silicon box. In *Proceedings of NASA/DoD Evolvable Hardware Workshop*, pages 167–176, 2002.
9. Adrian Stoica, Ricardo Salem Zebulum, and Didier Keymeulen. Mixtrinsic evolution. In *ICES*, pages 208–217, 2000.
10. A. Thompson. An evolved circuit, intrinsic in silicon, entwined with physics. In Tetsuya Higuchi, Masaya Iwata, and L. Weixin, editors, *Proc. 1st Int. Conf. on Evolvable Systems (ICES'96)*, volume 1259 of *LNCS*, pages 390–405. Springer-Verlag, 1997.
11. Stephen Wolfram. *A new kind of science.* Wolfram Media Inc., Champaign, Ilinois, US, United States, 2002.