Acknowledgements
Thanks to my supervisor Inman Harvey for some very useful comments.

Contents

1 Introduction

1.1 A Short Historical Perspective

(from ten light sensors spaced evenly around the robot (see below)) presented at each time phase.

Indeed, much of the impetus for research in this area comes from the firmly held belief that, as the tasks required of robots become increasingly more complex, it will be increasingly infeasible to hand-design robots to perform them [2] . In as much as, coupled with suitably chosen selective pressures on breeding (represented as a fitness function, often over time), a GA allows the problem to seek its own solution (the solution is, in this sense, emergent from the system created) the human programmer is freed from the task of designing the solution and can turn his attention to creating the right environmental pressures (finding the right fitness function) and the right encoding system (the way in which, for example in this case, the neural network is represented on the genome (see section below on encoding the network)) in which progressively better solutions will be found. This is, in itself, a difficult task, but is orders of magnitude simpler than having to hard-wire the robots.

Part I
Genetic Algorithms

3 Standard Methods Employed by Genetic Algorithms

3.1 Common Genetic Operators

3.1.1 Crossover

The most common form employed, crossover consists of combining elements from one genome with those from another to produce an offspring. The most common method is single-point crossover though multiple-point crossovers are sometimes employed. The method is as follows:

1. Take two parent genomes P1 and P2 of fixed length l and divide them at point x along the genome where x is a random number between 1 and l-1

```
                        |
P1   01110101101000101011|011000101001010
                        |
P2
```

3. Choose one of O1 and O2 to be the offspring of P1 and P2; in the GA I employ (see

optimisation. This proof was important because it showed, for these blocks, a remarkably large amount of search space could be combed in relatively few genetic operations on the genome population.

However, many areas of GA research defy mathematical formalism. There is much that is unpredictable and often heuristic, rules-of-thumb are all that is available to guide the development of new GAs.

The upshot of this is that much of GA research involves trial and error, a natural selectionist approach true to its subject matter. This trial and error is not, of course, totally uninformed; at the heart of most approaches is a putative gem of wisdom:

One approach currently gaining credibility is motivated by the belief that in restricting evolutionary forces to small pools or sub-populations within the main population of genomes, one may encourage diverse approaches to specific problems to flourish within the main population; this approach has been developed in response to a problem that consistently dogs the operation of GAs:

It is entirely possible that a genome which is initially relatively unfit may possess the key, or springboard, to a very fit creature -a good position in the n-dimensional search space- of the future. Equally, a genome which is initially fit may well prove redundant in the future, being further away in the search space from the optimum than the previous ailing genome. If both these creatures are encountered early on in the evolution of the main population and both are subjected to the full battery of evolutionary pressures -if both are in direct competition with one another- then it is probable that the latter will triumph at the expense of the former and a potentially useful future approach will be lost. If, however, we allow the less fit genome to breed away from the full rigours of the main population in a small pool then it may lead to the development of a better solution than the previously fitter genome. In short, it is recognised that encouraging diverse approaches to a problem is advantageous but that in GAs a fitter solution may well come to predominate too quickly, thus preventing any of the weaker but potentially useful approaches from reaping fruit. What is needed is an approach which compromises between a desire to see the fittest triumph and a belief that initially weak members of the main population should be protected from the full application of the Darwinian ethic; a time buying strategy to allow the different approaches to show their worth.

One way in which this can be achieved is by employing a spatialised GA. The algorithm is as follows:

1. Arrange initial n*n members on a toroidal (wrap around) array of dimension nxn.

2. To breed two creatures:

   (a) Choose a member of the main population (*parent1*) on a group-fitness preferential basis - where group-fitness is the average fitness of a member of the main population and his immediate neighbours- from a sample of n chosen at random; use rank-based-selection(RBS).

   (b) Restrict breeding to within the sub-population defined by *parent1* and its immediate neighbours.

   (c) Using RBS select a member of this subpopulation (*parent2*) to mate with *parent1.*

   (d) Crossbreed *parent1* and *parent2* and choose one of the crossed genomes to be the the *offspring.* Mutate this offspring.

(e) Choose a member of the sub-population using RBS biased in favour of the weaker members [5]. Replace this member with the *offspring.*

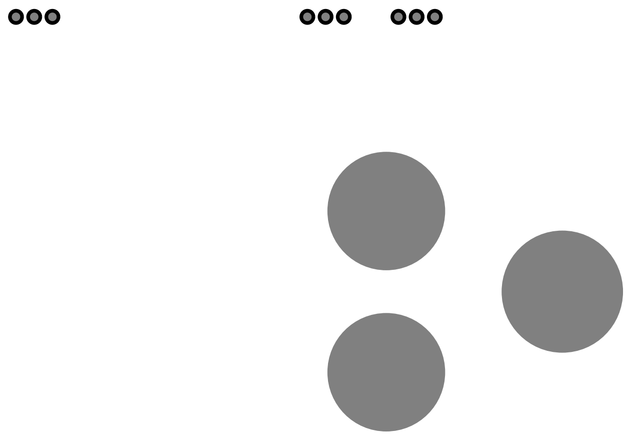
Breeding, within this sub-population is governed by rank-based-selection (RBS) (see section above) which is used to choose the breeding partners and the member of this sub-population to be replaced by their off-spring.

It should be recognised that these sub-populations merge together and that via crossover operation radically disrupts this context we might expect many of the parameters, post-crossover, to be distinctly sub-optimal; it is possible that two very fit robots, when bred, might lead to a very weak offspring.

This should not, however, provoke despair in the method of *strong direct-encoding*. Many weight connections and thresholds will probably be of some use in most networks. The weights and thresholds run in the range from -127 to +127. Negative values can be viewed as inhibitory (the node receiving input along a negatively weighted connection being less likely to register 1 as its activation value) while positive are excitatory (the node being more likely to register 1 as its activation value). Since the relative excitatory or inhibitory capacity of a connection or threshold is preserved through the genetic operation of crossover, and taking into account the fact that we might expect certain relative weights to be generally useful for producing a fit network, it is only necessary for the genetic algorithm to encourage these generally useful parameters for it to be successful; Holland's schema processing has shown that GAs do perform this function (see [1]) [8].

5.2 The Hybrid Encoding

While *strong direct-encoding* can be shown to work (see Results section below) its efficiency must be called into question. As has been mentioned, by pre-establishing the structure of the network we are already imposing what is almost certainly a sub-optimal state-space in which the GA is required to work; better structures for the task exist. Recent interest in evolving connectionist architectures [9] has focussed on using the GA to design structural features of the network, either with fixed weight and threshold parameters (typically binary



5.3 Search Space Consideration

The total size of the *strong direct-encoded* genome is 130 signed integers [15]. Parametrically this makes for a 130 dimensional state space.

In the *hybrid encoding* each output and hidden unit was allowed to define ten weighted connections. Although the total number of weighted connections was less (100 compared with 120) than the *strong direct-encoding*, because each connection required two numbers, (one to establish a link, the next to weight it) the search space, 210 dimensional, was larger than the *strong direct-encoding* method.

It was hoped that the general efficiency of genetic algorithms (see above), coupled with the freedom of the neural network to establish novel and better task-oriented network structures, would compensate for this increased search space and still allow the *hybrid system*, in the allotted 2000 breeding cycles, to outperform the *strong direct-encoded* method (see Results below for confirmation of this belief).

Part II
The Simulation

6 Practical Considerations

6.1 Availability and Speed of Processor

As a member of a networked system, processor time is shared out among a group and is as such fairly limited. The amount available affects considerations such as the complexity of the environment.

6.2 Designing the Environment

The environment in a simulated robotics project can be made arbitrarily complex. Processor intensive applications such as recursive ray-tracing (see section below) or complex kinematics can almost always be made more accurate. However, the availability of processor time meant that my environment had to be necessarily simplistic. The addition of generous quantities of noise should, however, alleviate to a certain extent the problem of unrealism (see section below on the importance of noise).

6.3 Choice of a Programming Language

Having been impressed by the speed of 'C' coded neural networks, I decided to teach myself this language and coded my programming project in it. It is not unusual for GAs to take days to achieve results -dependent, of course, on the problem posed- and the speed and efficiency of implementation is an extremely important factor.

7 he Environment

The environment in which the guard-robot finds itself is a two dimensional four-walled room (dimensions 400x400 units). The walls radiate light with an intensity of 1, whilst

[15](input-hidden) 10x6 + (hidden-hidden) 6x6 + (hidden-output) 4x6 + (hidden and output thresholds) 10.

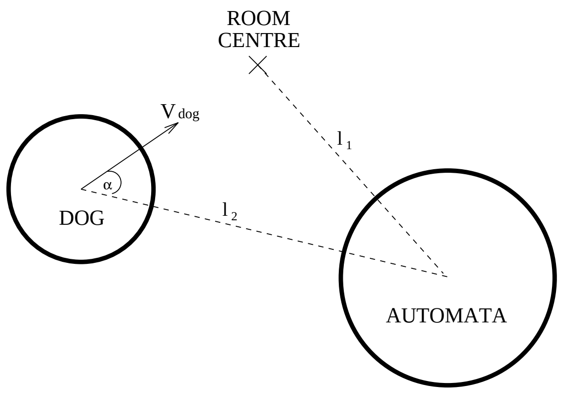


Figure 2: The Automaton's Movement

the automaton is black (intensity 0) (see section below on ray-tracing).

7.1 The Kinematics

The robot and automaton do not physically interact. This transparency saves processor time computing the collision

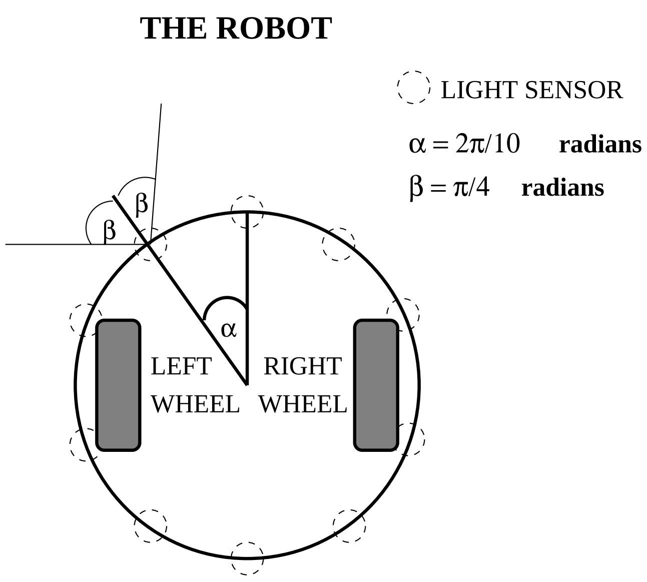


Figure 3: The Simulated Robot

(b) If $\cos(\alpha)$ is positive then the automaton is repelled along the line l_2 either by a force proportional to the robot's velocity times cos (α) divided by the length of l_2 or, in the case of the normalised automaton response (see below), at a speed of 60 units per time phase.

8 he Robot

The robot is circular (the environment being two dimensional) with ten light sensors arranged uniformly (every $(2\pi/10)$ radians) around its periphery. It's movement is effected by left and right wheels powered by separate motors (see Fig 3).

Information from the environment is obtained by the light sensors which return an average intensity rating over a defined arc-spread 2β (see Fig 3). Referring to Fig 4, normalised input from these (in the range $[0,1] \subset \mathbf{R}$) is sent directly to the ten input nodes of the robot's neural network [17] where the values are forward propagated through the network, according to weighted connections and node thresholds whose parameters are encoded on the robot's genome (see section on encoding the genome). This propagation establishes the activation values of four output units (in the range $[0,1] \subset \mathbf{R}$) O_1, O_2, O_3 and O_4. The left motor response is obtained by $O_1 - O_2$, the right by $O_3 - O_4$; this gives a possible range for each motor of between -1 and +1 (allowing forward and backward motion, and any discrepancy between the left and right motor outputs providing for a rotation motion) .

For the simulations run the left and right motor impulses at each time period were multiplied by the robot's maximum forwards and backwards velocity of 30 distance units per time period to give a distance over which each wheel travels; the robot's movement is resolved into a translation and rotation dependent upon the distance travelled by each

[17] analogous to its sensory nervous system, being employed to mediate between received sense data subsequent action.

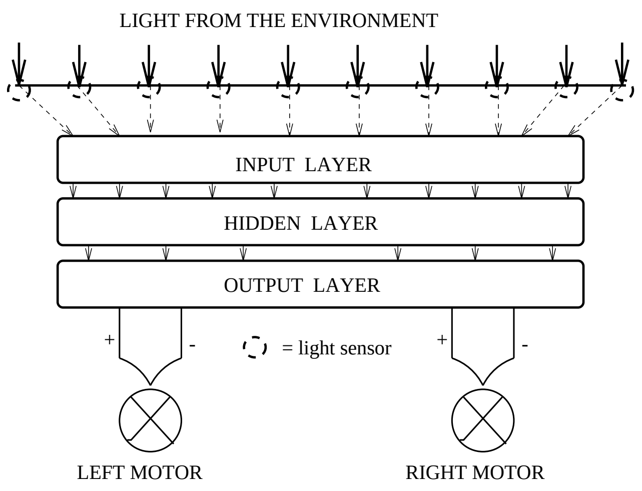


Figure 4: Converting Light Intensities to the Robot's Movement

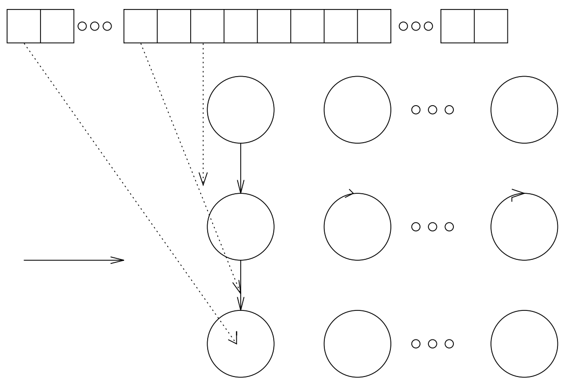
wheel.

Uniform noise at ±10% is added at all stages of the network's all weights and thresholds, can be said to partition this input space into discrete areas. Members of these areas provoke a similar output response which, applied to the motors, cause the robot to move in a set way; in this way, the robot can be said to possess a number of strategies. As the robot gets fitter, the partitioning of the input space into set output response comes to represent progressively better strategies for repelling the automaton from the centre; the guard-robot begins to develop a strategic understanding of the problem at hand. The distributive nature of the neural network; each unit participating in a large number of abstractions from the input space (for use in fulfilling useful discriminations of the environment) makes it an efficient way of storing the large amounts of knowledge the robot must employ if it is to develop a successful general approach [19].

The optimal solution to the task can be viewed as a vector in the parametric weight and threshold space. Successful solutions will tend towards this vector and the GA can be seen as the motive force for this tendency. In the case of the *hybrid encoding* scheme (see above) the state space is extended to include structural details about the net; the individual units are allowed to search for useful weighted links with other units, rather than having them pre-established.

9.2 The Neural Network

The neural network in the guard-robot is analogous to a central nervous system. It acts as an intermediary deciding, from a given sensory input -provided by the ten light sensors arranged around the robot- what action is to be taken by the robot -achieved through impulses sent to its left and right motor-driven wheels. The structure of the neural network is dependent on the encoding system used (see section on encoding the network).



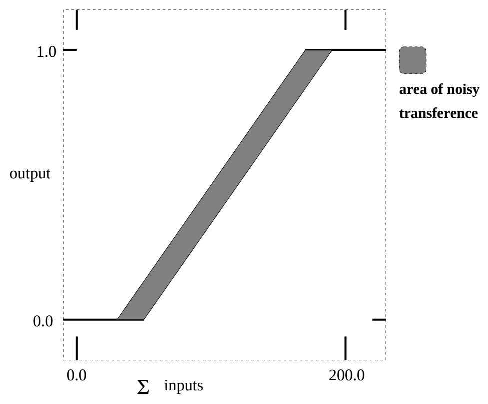


Figure 6:

matically formulated simulation which failed to take into account this ‘fuzziness’ in the real world would be of questionable relevance. The ability

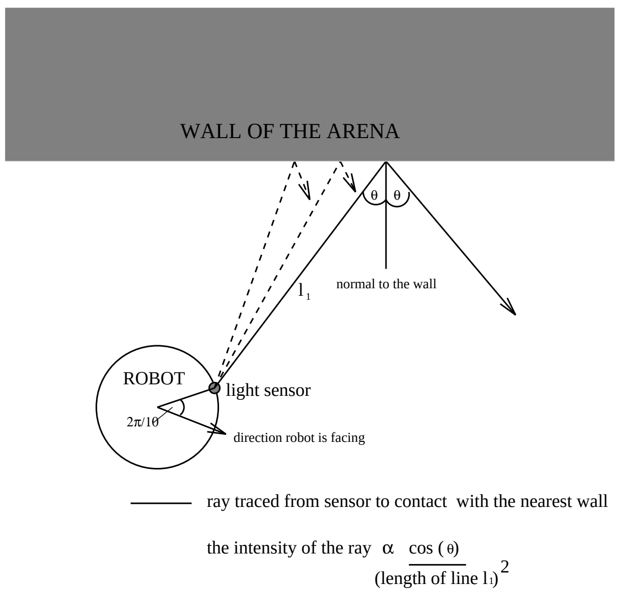

WALL OF THE ARENA
θ
θ
l_1
normal to the wall
ROBOT
light sensor
2π/10
direction robot is facing
ray traced from sensor to contact with the nearest wall
the intensity of the ray α cos (θ) / (length of line l_1)^2

Figure 7: The rays are projected from the light sensor and the intensity registered is a a product of their first encounter with an object (the automaton or a wall).

The simplified procedure I employed assumed that the light in the simulation radiated at an intensity of 1 from the arena walls. Each point on a wall could be assumed to be a point source of light directed at a normal to the wall. The automaton was assumed totally dark (any ray intersecting with its circle returning an intensity of 0) for maximum contrast with walls. The intensity value returned by a ray was calculated by taking into account only its first collision with an object.

In Fig 7 the ray leaves the sensor and intersects with the wall of the arena. The intensity registered is proportional to the cosine of the incident angle (θ) divided by the square of the distance from sensor to the point of intersection (the inverse square law for the radiation of light) times a constant c which regulates the distance from a wall at which a ray perpendicular to that wall will return the maximum intensity of 1. Using vector maths these values are calculated and an intensity for that ray returned. The intensity registered by the sensor is the average of the ten rays traced over the arc-angle.

The ray tracing employed was necessarily simplistic (see comments above on the importance of limiting the processor time necessary) but the general noisiness of the simulation - noise being injected at all phases of the operation of the neural network - does not require much more then a crude intensity indicator, and in this respect the procedure works well. Results obtained (see below) show that it provides enough information for the robot to successfully negotiate a guard-dog behaviour. Within the limits imposed by the available processor time it has been made as veracious as possible a simulation of light interaction.

following the multiple divergent rays this interaction might produce; each of these rays would in turn be followed and their progress recorded until their contribution (in terms of adjusting the eventual light intensity registered) was deemed insignificant, according to a pre-established threshold. The contribution

Part III

The Experiment

10 he Interface

A modest X-Windows interface was designed to allow monitoring of the GA's progress. This interface provided for the initiation and subsequent review of a GA. From the interface it is possible to initiate a GA and review its progress or that of an already evolved population (specified in the command line); monitoring of the behaviour of the fittest in the population, a graphical representation of the GA's performance and the spatialised fitnesses of the population.

11 he fitness function

The task at hand requires the robot to guard the centre of the arena from intrusion. The further away from the centre the automaton can be kept, the better the robot is doing its job. I decided to employ a Gaussian function ($\mathcal{G}$) giving an optimal fitness per unit time of one if the automaton is confined to the corner of the arena (as far from the centre as possible) and approximately zero (using a suitable radius of Gaussian) if the automaton is at the centre:

$$\mathcal{G} = e^{\frac{-r^2}{c}}$$

where r is the distance of the automaton from the centre and c is a constant chosen to ensure a fitness return of approximately 0 for a centralised automaton.

Fitness is added at each time cycle giving an optimal fitness, for 400 time units, of 400. However, speed constraints and the general noisiness of the environment mediate against fitnesses of over 200. An optimum fitness give the relative severity of the fitness function (its non linearity means that maintaining the automaton at around half the optimal distance from the centre accrues far less than half the optimal fitness) is around this 200 mark.

12 Preliminary Details

A number of important parameters in the experiment are not under the control of the GA. Possibly the most important is the arc angle over which each light receptor collects light. I chose 90 degrees as a reasonable compromise (in Fig 3 angle β set at 45 degrees) and though this is probably sub-optimal to the task (for which possibly the most important element is discriminating the automaton from the background) it remained constant so that comparisons as to the effectiveness of the different encoding schemes could be made.

The starting positions for the robot and automaton are shown in Fig 8. The automaton starts at the centre with the robot placed such that the automaton falls just outside its radius of influence. At time t with the automaton at the centre, the fitness function (see Section) returns approximately 0, so in the number of time steps allocated (400 was chosen since this meant the 2000 or so breeding cycles could be completed in a realistic time) the robot's centre must enter into the circle of radius 100 (its radius of influence) from the

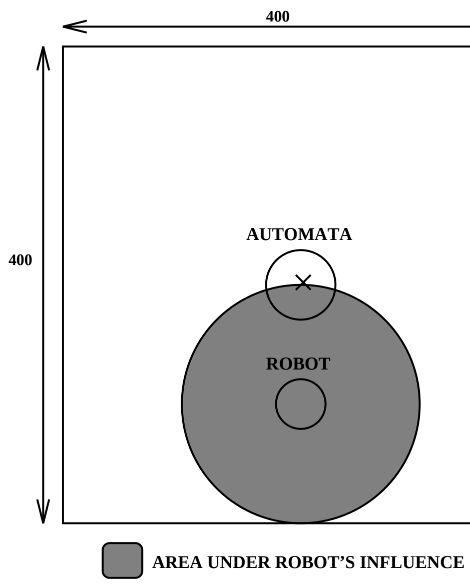


Figure 8: The Initial Positions in the Arena

centre of the arena if it is to effect the automaton and register a significant fitness for the trial. This discriminates heavily in favour of robots attracted to the automaton or with a tendency to centralise themselves in the arena -both good strategies.

13 Calculating the Fitness of an Offspring

An offspring is chosen by applying the algorithm for a spatialised GA (see section above on choosing a GA). The fitness of the offspring is then decided by giving it a number of trial runs in the arena:

13.1 The Trial Based system

The starting position for each trial is as described above but the initial orientation of the robot is random (a value in radians in the range $[0,2*\pi] \subset \mathbf{R}$) to encourage durable guard-dog behaviour; a robot which just moved forwards regardless of input might do well every now and then but could not be said to be guarding anything.

radians

at the centre of the arena whilst turning very quickly proved a very effective strategy and the fittest of the early simulations all possessed a variant on this strategy, with marginally different sizes of epicycle and speeds (see Results below).

In order to encourage more varying behaviours, the decision was taken to normalise the speed of the automaton; its repulsion being constant regardless of the speed of the robot.

By mediating against the quick epicyclists the hope was to encourage behaviour which

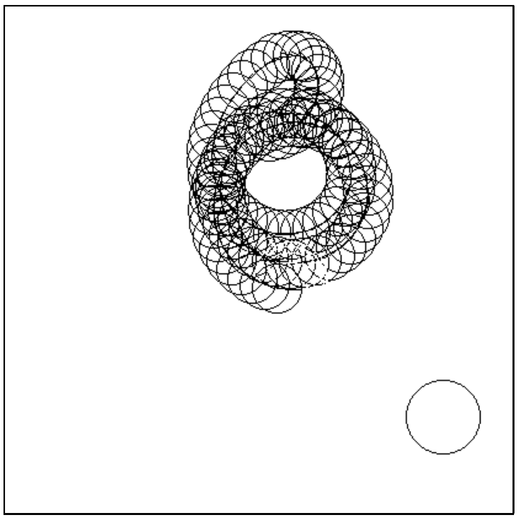


Figure 9: The *epicyclist* makes a sharp turn away from the wall and towards the centre of the arena

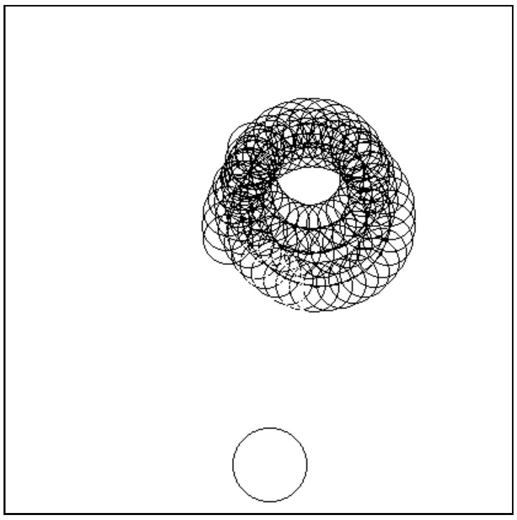


Figure 10: The *epicyclist* begins facing the centre of the arena and proceeds to colonise it

proximity, still generate enough repulsion to score a high fitness, now robots maintaining a closer proximity (within 100 distance units) to the automaton are guaranteed more repulsion.

Normalising the repulsion of the automaton proved surprisingly effective at producing varying robot behaviours (see Fig 17 and Fig 19). Optimising speed in a fixed position no longer proves necessarily the best strategy for the robot. While fast robots were observed (see *curly robot* in Fig 9 and Fig 10), speed now appears to be employed more as a means of influencing large areas of the arena in a short space of time [23] whereas before this useful strategic employment of speed was subsumed by the general efficacy of speed as a means of generating proportionate repulsion.

One of the most successful robots bred, I nicknamed the *big dipper* (see below for an analysis). Eschewing the large ellipses employed by *curly robot*, the *big dipper* employs a more direct approach. Its general behaviour is shown in Fig 17. Circling for a few time cycles it then moves rapidly to colonise the centre of the arena. This colonisation is more localised than the previous epicyclical strategies, relying on the robot's continuous presence at the centre, rather than its speed, to score high fitness points (see below for a fuller analysis of the *big dipper*).

15.3 Direct Versus Hybrid Encoding

The availability of processor time prevented a large number of trials being run, but for the purposes of this project three to four trials are assumed representative.

With the parameters fixed such that a comparison could be made [24] simulations were run and relative fitnesses compared.

[23]the larger the epicycles employed by the robot, the more likely that, at some point, the automaton will evade the robot's influence and return to the centre, thus costing the robot valuable fitness points. One way to compensate for this is to increase the speed of the robot.

[24]the light sensors averaging over 90 degrees; the repulsion normalised.

15.3.1 Degrees of Convergence

Of interest in monitoring the performance of the genome populations was the degree of convergence exhibited; the extent to which the fitnesses in a population are similar to one another; a guide to this is how the best performance for the population is reflected in the

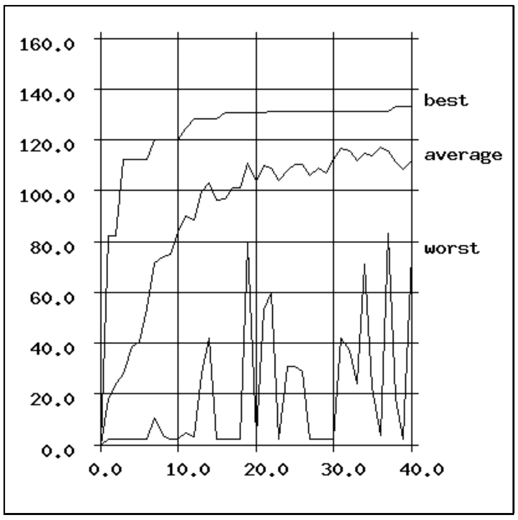


Figure 13: A typical performance by the *hybrid encoding* scheme. This simulation bred the

The *Big*

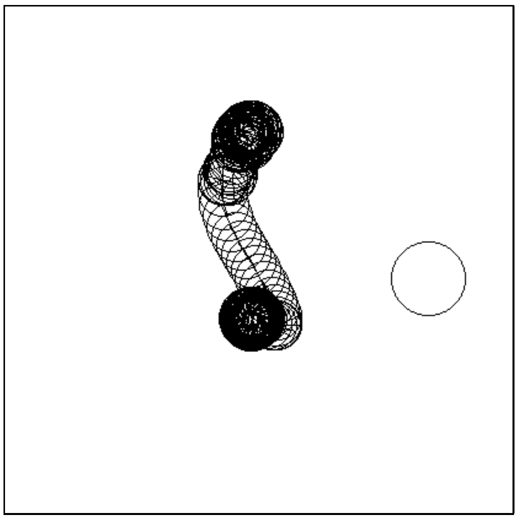


Figure 17: The performance of the *big dipper* with noise injected

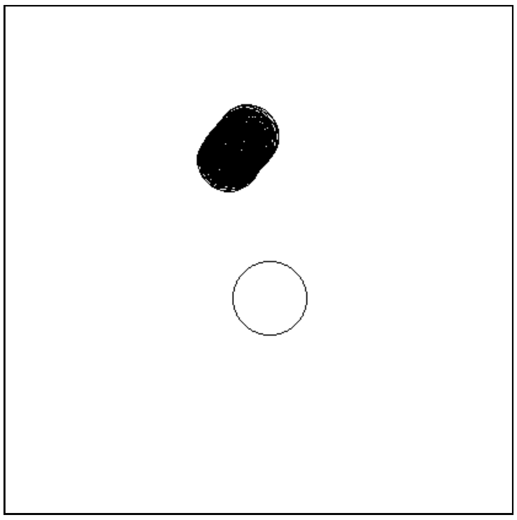


Figure 18: The performance of the *big dipper* with no noise added to the system

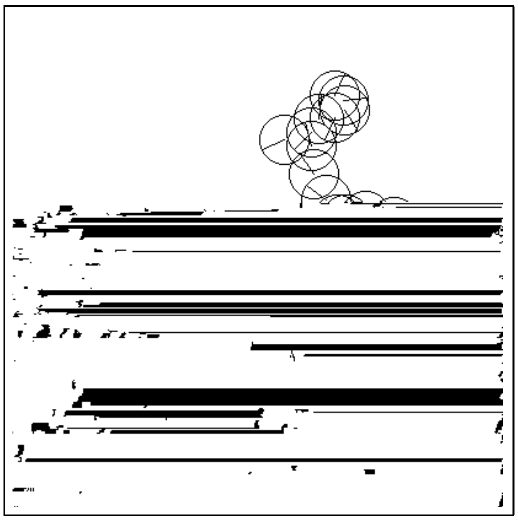


Figure 19: The *corkscrew* traps the automaton in the corner

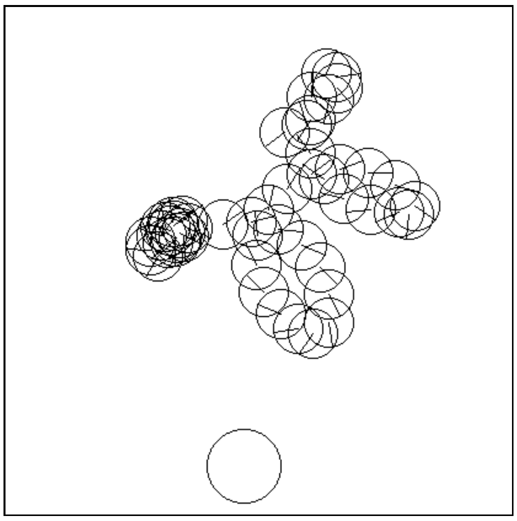


Figure 20: A standard response by the *corkscrew*. Notice the colonising behaviour, shown by the dark circle

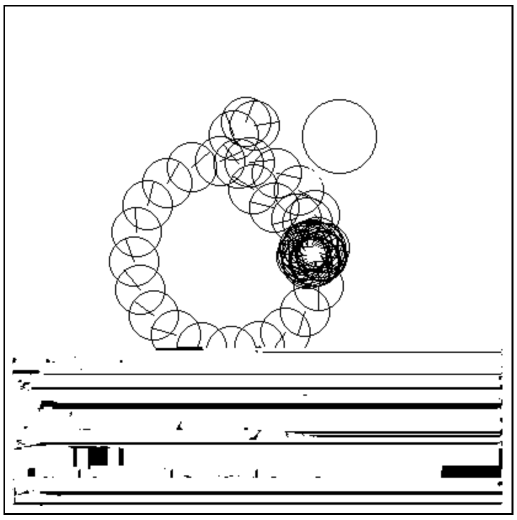


Figure 21: The *corkscrew* colonises the centre

15.4.2 Analysing Motor Response

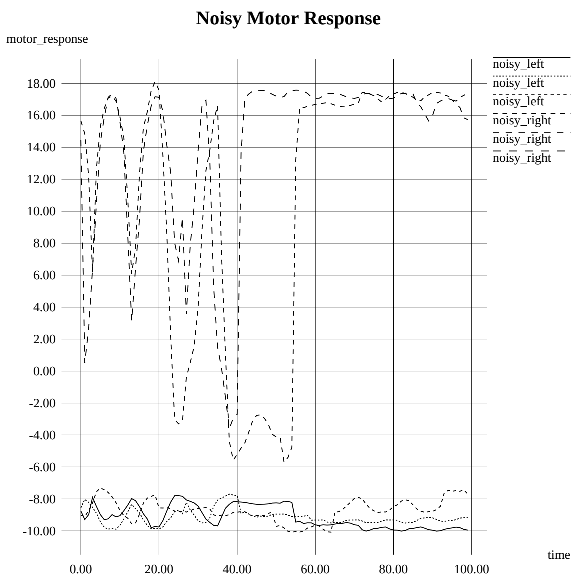
The *Big Dipper*

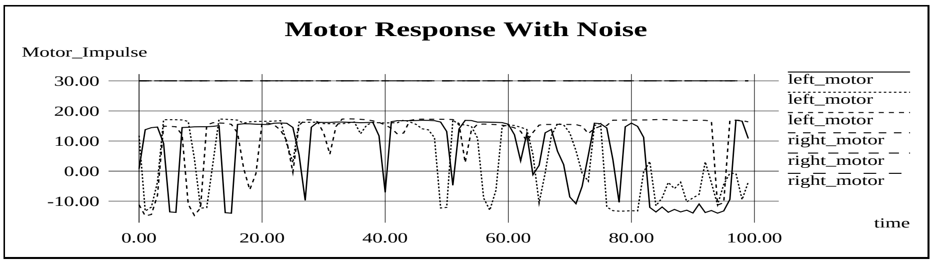
In Fig 22 the left and right motor response of the *big dipper*, with noise added, is shown over the first 100 time units of its progression. Its standard motion is characterised above and can be seen in Fig 17. The left motor response remains low throughout the 100 time units, though varying slightly. However, the right motor, normally with a high positive impulse, is characterised by occasional periods of low activity where the graph is seen to dip. Three consecutive trials are superimposed and this dip is a regular feature of all three. During the time when its right motor output is low, the rotational component of the robot's motion is minimised and its translational component increased [26]; in other words it stops spinning and covers a relatively large amount of ground. The sudden spurt towards the centre that characterises the *big dipper's* movement is explained by this sudden diminution in the right motor output.

Fig 23 shows the motor responses of the *big dipper* without noise. The right motor shows a high periodic response which keeps within a fairly small band. The left motor has a small response but it is periodic. This response fits the observed rotational behaviour (see Fig 18). Unlike that in Fig 22 the right motor response does not occasionally dip, allowing for increased translational motion; the noiseless response is that of a robot with a consistently high rotational component of motion and very little translation.

The part noise plays in allowing the *big dipper* to move to the centre seems clear. Recurrencies within the network build up and occasionally are enough to push the robot into making a positive move. This positive action is required to move the robot from an ambiguous position to a more certain one; once it has begun to move towards the centre, interactions with the automaton become increasingly likely and ambiguity diminishes.

Looking at Fig 23 the right motor output oscillates periodically. It is probable that the sudden dip in the noisy motor response (see Fig 224lopositdden


Noisy Motor Response
motor_response
18.00
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
-2.00
-4.00
-6.00
-8.00
-10.00
0.00
20.00
40.00
60.00
80.00
100.00
time
noisy_left
noisy_left
noisy_left
noisy_right
noisy_right
noisy_right



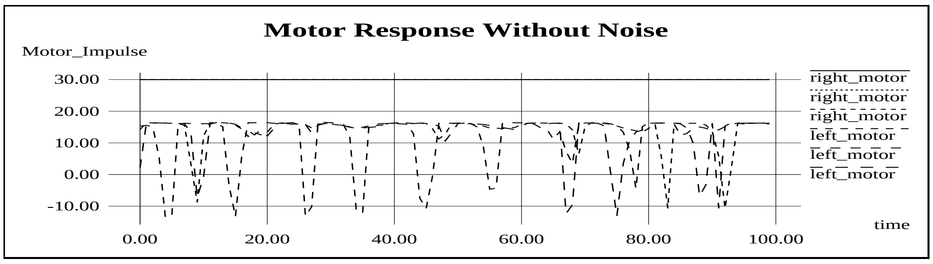


Figure 24: The Motor Response of the robot *corkscrew* with and without noise; three consecutive performances over 100ormanc3.79l1990.747573.09l2010.463272.2376066(ormanc3.c69p00100Tsi

Nevertheless, a number of parameters in the simulation were under my control that I feel it would have been better to submit to environmental pressures. For example, in the *hybrid encoding* system the number of connections each unit could receive was pre-established. It would doubtless be far more efficient, and in the long run effective, to allow the GA to establish these parameters [28]. I feel now that I allowed too many weight connections; the larger weight space this involves required more breeding cycles than time allowed for to do it justice.

Though the results achieved with the *hybrid method* were superior to those managed by the *strong direct-encoding* (see Results above), areas such as the number of weight connections allowed to each unit, the possibility of allowing a unit to send impulses to another rather than just receive them (though this would require an extra indicator on the genome), the number of cycles the neural network propagates during each time unit etc., suggest improvements for the future.

17 Conclusion

The genetic algorithm employed proved successful in breeding network controllers that simulated a guard-dog behaviour in the robot. The fitness of the population increased progressively for around 1500 breeding cycles and then was then seen to fall off as further progress became increasingly unlikely.

Though the environment simulated was necessarily simplistic, this does not invalidate the results achieved. Within the large search space defined, the GA was able to breed neural network controllers that enabled the robot to perform its specified guard-behaviour with some efficacy. The task is of suitable complexity that I feel a human trying to design an intermediary, between the information from the robot's light sensors and its subsequent response, that performed the task as well would be faced with a considerable challenge; certainly one greater than designing a spatialised GA to breed an attempted solution.

While this project has shown that the approach of breeding network controllers has some efficacy, the success of the *hybrid encoding* scheme (see Results) over the conventional *strong direct-encoding* approach highlights the ever-present possibilities for innovation. The more parameters left under the effective pressures of natural selection, the better the results seem to be. This seems in validation of the belief that, in some areas, natural selection is a more effective tool than human intuition. An obvious next step would be to place as many parameters as possible under the control of the GA, towards the aim of limiting, as far as possible, any rigid structural assumptions imposed from above [29] ; my intuition is that the freer the GA to explore the more impressive will be its eventual solution.

The time available meant that a control experiment, using a more conventional GA (in this context, non-spatialised) could not be run, and, as such, no figures are available for comparing relative performance. It is only speculation, but the fact that diverse approaches to the problem (see Results) did surface leads me to believe that the spatialised GA fulfilled at least this aspect of its remit.

[28]in [4] an example is given of a variable length genome; the genome being extendible allows it to increase its own search space. This makes regulating genetic operations a more complex operation, but the advantages in terms of flexibility seem to outweigh this aspect.

[29]However, as mentioned, care must be taken not to expand the search space too much. The incremental approach favoured by Harvey, Husbands and Cliff (see [2],[4] and [9]) where the genome is of variable length and able to increase its own search space points towards a possible solution to the problem of initially overloading the genome.

REFERENCES

[1] Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence.* Ann Arbor, MI: University of Michigan Press, 1975.

[2] I. Harvey, P.Husbands and D.T. Cliff. Issues in evolutionary robotics. In J.arTd[(A)199(,tics.)]Te.959