Co-evolution of Pursuit and Evasion II: Simulation Methods and Results*

Dave Cliff[1] **and Geoffrey F. Miller**[2]
[1]School of Cognitive and Computing Sciences
University of Sussex, BRIGHTON BN1 9QH, U.K.
davec@cogs.susx.ac.uk
[2]Center for Adaptive Behavior and Cognition
Max Planck Institute for Psychological Research
24 Leopoldstrasse, 80802 MUNICH, Germany.
Miller@mpipf-muenchen.mpg.de

Abstract

In a previous SAB paper [10], we presented the scientific rationale for simulating the coevolution of pursuit and evasion strategies. Here, we present an overview of our simulation methods and some results. Our most notable results are as follows. First, co-evolution works to produce good pursuers and good evaders through a pure bootstrapping process, but both types are rather specially adapted to their opponents' current counter-strategies. Second, eyes and brains can also co-evolve within each simulated species – for example, pursuers usually evolved eyes on the front of their bodies (like cheetahs), while evaders usually evolved eyes pointing sideways or even backwards (like gazelles). Third, both kinds of coevolution are promoted by allowing spatially distributed populations, gene duplication, and an explicitly spatial morphogenesis program for eyes and brains that allows bilateral symmetry. The paper concludes by discussing some possible applications of simulated pursuit-evasion coevolution in biology and entertainment.

1 Introduction

At SAB94 we argued that it's important, interesting, and useful to simulate the co-evolution of pursuit and evasion strategies, and presented some preliminary results [10]. In that paper we presented a review of the relevant literature (e.g. [7]); since then, other comparable work has appeared (e.g. [13, 14]).

In the intervening two years of project development, we have improved our methods and generated more detailed and more complete results. This paper gives an overview of our current methods and some results from this ongoing research. We include only the briefest recap of the intellectual motivations and scientific background for this project, which we have covered at length elsewhere [9, 10, 11, 6]. Much of our effort in this project was directed at improving our simulation methods to cope with the sometimes frustratingly slow and baffling process of coevolution. The populations were changed from traditional "pools" to spatially organized matrices with local competition, mating, and replacement. Many evaluation functions were tried and failed before a few were found that worked. We now use a more efficient contest method where individuals play the "Last Elite Opponent" (LEO) [14, 12]. New analysis methods were developed to see whether directional progress is occurring during coevolution, including cross-generational bit-string correlation matrices and gene persistence plots (see [6]). This paper is offered in part as a traveler's advisory about the pitfalls of venturing unprepared into the Red Queen's protean world.

In brief, the intellectual background for our work stems from the observation that animals survive and reproduce by exploiting "fitness affordances" [11] in their environment, and that some of these affordances are themselves mobile animals that do not want to be exploited (e.g. prey, coy females, hosts avoiding parasites). This basic conflict of interest between two mobile animals typically leads to a pursuit-evasion contest, where one animal tries to catch the other to do something to it (eat it, mate with it, suck its blood). Over generations, these contests result in a co-evolutionary arms race between pursuit strategies and evasion strategies, where animals evolve to be faster, more maneuverable, better at predicting each other's next moves, and better at being unpredictable [10, 9]. Pursuit and evasion have been studied from different angles by behavioral biology, neuroethology, differential game theory, and previous SAB work. Thus, pursuit-evasion contests are among the most important, challenging, and co-evolutionary of all animate behaviors. But in their scientific analysis, there is a large gap between the over-simplicity of game-theory models and the baffling complexity of real pursuit-evasion wetware as studied by neuroethology. SAB-style simulation may help fill the gap, by illuminating the co-evolution of strategies that are complex enough to include interesting examples of perceptual specializations, prediction, and proteanism, but not too complex to analyze.

*To appear in: P. Maes, M. Mataric, J.-A. Meyer, J. Pollack and S. W. Wilson (eds) *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior (SAB96).* MIT Press Bradford Books 1996.

2 Simulation Methods

Our pursuit-evasion coevolution simulator has been developed and refined to serve the study of co-evolutionary arms races, to characterize the resulting behavioral strategies, and to analyze the underlying sensory-motor architectures that generate the observable behaviors through interaction with the environment. The simulation approximates a spatiotemporally continuous system by updating a model of the animats in their environment at a rate of 100 frames per simulated second. Here we summarize some of the most significant details.

The following sections describe the equations governing the physical dynamics of motion for the animats, the artificial neurons used in the simulations, the genetic encoding of the controller specifications as bit-strings for operation on by the genetic algorithm, and then details of the genetic algorithm itself.

2.1 Physical Dynamics

In all our work to date, we have employed a two-dimensional (2-D) simulation, where circular animats chase each other around on an empty infinite plane: there are no obstacles or boundaries in the animats' world. The primary motivation for such a gross simplifying assumption is one of computational economy. Despite the lack of a third spatial dimension, we employed a fairly realistic model physics.

The neural network for each animat gives two output values referred to as v_l and v_r: these are treated respectively as left and right motor signals, giving a differential-steer system (i.e. the rate of change of angle of orientation is dependent on the difference between the left and right motor settings, as in many two-drive-wheeled mobile robots). The equations of motion are based on simple Newtonian point physics.

Specifically, the relevant equations for translating from the output values to the accelerations are:

$$m\frac{d^2 f}{dt^2} + c_f\frac{df}{dt} = k_f\frac{v_l + v_r}{2} \tag{1}$$

for forward movement, where m is the animat mass, c_f is a friction coefficient, and k_f is a scale factor; and:

$$\vartheta\frac{d^2\alpha}{dt^2} + c_a\frac{d\alpha}{dt} = k_a(v_l - v_r) \tag{2}$$

for changes of orientation angle α, where ϑ is the animat moment of inertia, c_a is an angular friction coefficient, and k_a is a scale factor.

The use of these 'realistic' equations of motion mean that the animats take time to accelerate to a desired speed (either linear or angular). Note also that there are no 'brakes': rather, the drag/friction coefficients mean that if there is a reduction in motor output, the speed falls exponentially towards the new steady-state value.

The values of $m, c_f, k_f, \vartheta, c_a$, and k_a are held fixed for any one experiment, but we have experimented with a number of settings of these parameters. Clearly, the relative settings can have a large effect on what constitutes a sensible strategy for pursuit or evasion. In the limit, the relative settings can determine the likelihood of anything interesting happening. Figure 1 shows an impressionistic rendering of the effects the relative pursuer/evader dynamics have on the outcome of the contests. A schematic 2-dimensional space is shown, with the horizontal axis being a measure of the angular motion capabilities of the pursuer (P) relative to the evader (E), and the vertical axis measuring the relative linear motion capabilities. In the horizontal, there will be some threshold value past which the pursuer can reliably out-turn the evader, and in the vertical there will be a threshold beyond which the pursuer can in principle catch up with the evader. These two threshold lines partition the space into four zones. The further into the upper-right zone, the greater the chances of pursuers winning all contests (they can catch up with the evaders, and out turn them too); the further to the lower-left, the more likely the evaders are to do very well (the pursuers can neither catch them nor out-turn them). Thus, if there are asymmetries in the dynamics (i.e. the pursuers and evaders have different parameters for the equations of motion) then for interesting trials, the pursuer's maximum linear speed should be greater than the evader's (so the pursuer can, in principle, catch the evader), and the evader's maximum angular speed should be greater than the pursuer's (so the evader can, in principle, dodge and out-turn the pursuer).

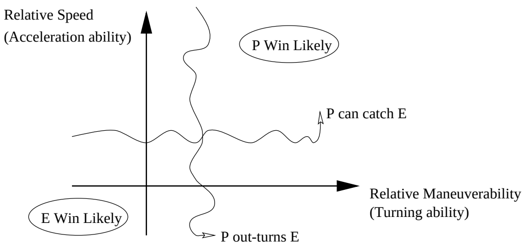

Figure 1: Impressionistic illustration of the effects of relative kinematics on the likely outcome of a pursuit-evasion contest. See text for explanation

The results described later in this paper come from experiments where the kinematics were *symmetric*: that is, the pursuers and evaders had identical parameter values for their equations of motion. Thus, the pursuer and evader have the same accelerations and maximum speeds, which gives a clear advantage to the evader. This is because, as long as the trial starts with sufficient distance between the pursuer and evader for the evader to turn to face away from the pursuer before the pursuer can hit the evader, the evader need only perform such a turn and then accelerate to top speed to avoid being caught – its top speed is the same as the pursuer's. To redress the balance, the animats were given a limited stock of 'energy' or 'fuel', with the pursuer having (slightly) more initial energy than the evader. Energy consumption rose as a quadratic function of linear force exerted, so an animat could 'dawdle' by moving slowly for a long period of time, or 'burn out' by accelerating to maximum speed and using up all its energy in a few seconds. When energy fell to zero, the motor outputs were disabled and the animat would drift to a halt, being slowed by the simulated friction. Small amounts of random noise were also added onto the values of v_l and v_r, so that movement was nondeterministic.

2.2 The Neuron Model

We used continuous-time recurrent neural networks of a type which have recently been the subject of detailed analysis (e.g. [1]). The activity of a single 'neuron' processing unit is described by the equation:

$$\tau_i \frac{dy_i}{dt} = -y_i + \sum_{\forall j} w_{ji}\sigma_j(y_j) + \mathcal{N}_i(t) + \mathcal{I}_i(t) \qquad (3)$$

Where: y_i is the activity of unit i; τ_i is the time-constant for unit i; $\sigma_j(\xi)$ is a sigmoidal function with bias term (threshold) θ_j; w_{ji} is the weight of the connection from unit j to unit i; $\mathcal{N}_i(t)$ is spontaneous noise injected at time t, from a uniform distribution centered on zero; and $\mathcal{I}_i(t)$ is external (sensory) input at time t.

For each unit, the values for θ_i and τ_i, and the upper and lower bounds on the distribution of $\mathcal{N}_i$ explicitly set by a sequence of bits on the genotype. The values for w_{ji} and $\mathcal{I}_i$ were also genetically specified, but not explicitly. Rather, they were derived from the 'morphogenesis' process: this is described in Section 2.3. The connection strengths w_{ji} remained constant throughout the 'lifetime' of each animat: in this sense, there was no opportunity for 'learning'.

We had only one sensory modality: simulated 'flatland vision' in the 2D plane. Photoreceptor cells could be placed on the perimeter of the animat's circular body, with their location, orientation relative to the body, and angle of acceptance all being genetically specified. The response of a photoreceptor was proportional to the percentage of its acceptance angle that was *not* occupied by the opponent's circular body. The relevant 2-D projection equations are trivial when it is known that there is only one object in the visual world. The equations could thus be solved analytically, without recourse to numeric approximation techniques; but random noise was added in to ensure that sensing was nondeterministic and that there was a lower limit on resolution (i.e. small signals could be swamped by noise).

2.3 Genetic Encoding and Morphogenesis

The architectures for the neural-network sensory-motor controllers were specified by bit-string genotypes. In common with past work in evolving such controllers for artificial autonomous agents (e.g. [4]) we treat the arrangement of the sensory morphology as an intrinsic part of the controller-network specification. The controller-network is encoded as strings of binary digits (bits). Rather than use variable-length genotypes to allow varying numbers of units in the networks (e.g. [4]), we use fixed-length genotypes, partitioned into a number of 'fields'. At the start of each field is a sequence of bits which governs whether that field is 'expressed' – i.e. whether it is ignored or read to form part of the specification for the network. Each field contains bit-sequences that govern parameters that define an individual neuron: the neuron has a particular genetically-specified physical location within the animat's 2D circular body; and there are genetically set parameters that determine the growth (on the body) of a fractal 'input' tree and a fractal 'output' tree. If the input tree of one unit intersects the output tree of another unit, and certain other genetically specified conditions are met, then a connection is made joining the input to the output, with the strength of the connection being affected by the geometry of the intersection of the two trees. If the unit's output tree terminates within a central zone on the animat's body, then it becomes a motor output unit, affecting the left or right output value depending on the precise geometry of the termination point(s). If a unit's input tree extends beyond the perimeter of the body, then that unit becomes a visual input unit, with the location, orientation, and acceptance-angle of the corresponding photosensor being set by the geometry of the tree's intersection with the body-edge. Finally, each field contains a sequence of bits that can specify *symmetric* expression: if a neuron is expressed symmetrically, then a copy is created by reflecting the neuron's body and input and output trees about the animat's longitudinal axis. Full details of the encoding scheme are given in [3].

This encoding scheme has proven to offer considerable power. It allows for some units in the network to be "hidden"(cf. interneurons), others to be sensory input neurons, and others to be motor output neurons. But most interestingly, it is possible for a unit to be *both* sensory *and* motor. This capability, coupled with the relative ease of generating bilaterally symmetric designs, means that random genotypes often encode simple but very effective Braitenberg-*Vehicle*-like controller architectures (cf. [2]). Almost equally as often, random designs may have no motor outputs, no sensory inputs, or no connection between sensors and motors.

For this reason, we use "animat eugenics" to increase efficiency: when generating the initial random population of genotypes, or breeding a new individual from two parents, a viability check is made: individuals with insufficient numbers of sensors or motors are rejected immediately. Then the genome is expressed to give a network, and a garbage collector deletes any disconnected units. If, after garbage collection, the network is still viable, it is added to the population. Otherwise, the genotype is discarded and another new one generated and tested: the process repeats until a viable genotype is generated.

One aspect of using this encoding scheme in practice is that typically there are significant sequences of bits on the genotype which constitute 'junk DNA', in that they don't contribute to the phenotype. Bits in a field which has the expression sequence set to "off" have no effect on the phenotype, and nor do bits that specify neurons that *are* expressed but subsequently deleted by the garbage collector because they don't connect to anything. However, we are cautious about the use of the term 'junk DNA': subsequent recombination or mutation could lead to an unexpressed field being expressed or to a disconnected neuron being connected. Thus, we prefer use of terms such as 'silent' or 'resting' DNA rather than 'junk'. One practical effect of silent DNA in the genomes is that a mutation in a silent sequence has no immediate effect. To counter this, we use mutation rates which would be considered very high in more standard encodings where every bit counts.

2.4 The Genetic Algorithm

Our genetic algorithm (GA) involves a population of pursuers and a population of evaders. Both populations are the same size. We use a spatially distributed GA, where individual genotypes in each population are spread out to occupy discrete positions on a grid. Individuals will only breed with nearby neighbors, and the offspring will replace nearby less-fit individuals. The grid has a toroidal topology so there are no edge effects.

The GA proceeds in discrete generations. In the very first generation, each individual is tested against a set of randomly-chosen individuals from the opponent generation. In all subsequent generations, individuals are tested against the best individual in the opponent generation on the previous generation – this "last elite opponent" (LEO) evaluation was introduced in [14, 12].

In each test, the genotype is given a number of *trials*: these are individual pursuit-evasion contests. The initial conditions for the trials are varied using standard statistical precautions. The final fitness of the individual was calculated as the mean fitness score from the set of trials. The evader fitness score was simply the length of time it lasted before being hit by the pursuer. The pursuer's fitness score was slightly more complex: they received fitness points if they were approaching the evader, and received a bonus if they hit the evader; this bonus was dependent on the time of the collision, and was two orders of magnitude bigger than the points received for approaching the evader, so pursuers that collided with evaders were much more fit than those which stealthily tracked their opponents. It should be noted that *many* fitness functions were tried before we arrived at this successful combination.

In each trial, the pursuer and evader start at their initial positions and orientations, with all activities in their neural networks set to zero. Once the trial has started, it continues until the animats collide, or both run out of energy and drift to a halt, or a fixed timelimit is reached.

Once all individuals in both populations have been evaluated, two new populations are bred. The breeding uses the standard GA operators of mutation and crossover, plus a duplication operator discussed further in Section 3.1. To breed a new individual, two 'parent' individuals are chosen using probabalistic rank-based selection. One of the two parents is chosen at random, and the process of copying its genotype into the 'child''s genotype commences. Once each bit is copied, a random number is generated from a uniform distribution. If this is less than some threshold value, then the copying pointer is switched to the corresponding position on the other parent. The number of crossovers in the reproduction of any one 'child' thus follows a poisson distribution: there will always be a finite probability that reproduction is asexual (i.e. number of crossovers is zero). Mutation is also applied on a bit-wise basis, with a different threshold value, so the number of mutations per reproduction also follows a poisson distribution.

3 Results

As is common in our experience of using artificial evolution, we witnessed many failed experiments before we experienced consistent evolution of animats exhibiting desired types of behaviors. Here we make a general observation about what we learned from one of the causes of our many failures, and then give a behavioral-level overview of a run that yielded some interesting results.

3.1 A Lesson from Failures: Duplication

The genetic encoding and morphogenesis technique we employed, along with the viability checks and garbage collection, almost guaranteed that at the start of a run, the most successful individuals would have very simple neural architectures similar to Braitenberg's *Vehicles*; with a symmetrical arrangement of a few (typically two or four) sensors connected to the motors by either 'crossed' or 'uncrossed' connections (i.e. connected contralaterally or ipsilaterally). Very often, any subsequent improvement in performance was due solely to variation and selection operating on the parameters of the individual neurons in the animat: the number of neurons almost never increased, even when there was a capacity on the genotype for ten or fifteen more. The improvement in performance, even after 1000 generations, was often only a small percentage of the initial performance of the random-genotype architecture.

We came to understand that this was because these small Braitenberg-style controller networks typically have very high epistatic boundaries around them in genotype space. That is, adding in one or more extra neurons *at random* almost always resulted in significant degradation of performance. The initial networks were so small that adding new neurons typically has a huge negative impact on the network's performance.

To remedy this, we borrowed another genetic operator from nature. Whereas many artificial evolution systems employ crossover and mutation, *duplication* is somewhat less widely used. In our implementation, duplication would take a single genetic field and copy it into another field on the genotype. If the field had its expression bits set to an active sequence, the resulting controller network would have extra neurons, but they would be (almost) identical to the units in the current successful network, and so would be unlikely to have a negative effect on the overall behavior. Subsequent mutations could smoothly alter the parameters encoded on one of the copied neuron fields, allowing for the other copy to either remain the same or, more importantly, to change and hence coadapt to the effects of the new unit(s). Once duplication was employed, we found that the number of units would often increase significantly over the course of a run, and this would be reflected in significant increases in performance. The discussion of Figure 10 in the next section illustrates duplication in action.

3.2 A Run That Worked

Results from individual trials between the elite pursuer and elite evader at various stages in a co-evolutionary experiment are shown in Figures 2 to 9. These figures all come from the same single run with pursuer and evader population sizes of 100, arranged in 10×10 grids, lasting for 1000 generations. We used LEO evaluation, with 15 trials per test; fitness being the average score. The time-limit on each trial was 15 simulated seconds, which typically took less than one second of real time. Animats were viable if they had at least two motor output neurons and at least one sensory neuron. The genetic algorithm was configured to give (on the average) 1.1 crossovers, 7.0 mutations, and 0.1 duplications per reproduction event. The genotypes had seven unit-fields, so the maximum number of units in an animat's network is 14 (i.e. all 7 fields expressed and bilaterally symmetric).

Figure 2 shows traces of the 2D paths made by the elite pursuer and evader from generation 0. These two animats are the best of the initial population, with randomly-specified genotypes. Traces of some significant variables are shown in Figure 3: the first nine graphs are, from the top: distance between animats (in meters: the animat bodies have a radius of 9mm); pursuer's 'target bearing' (in radians: a value of zero implies that the evader is straight ahead); evader's 'target bearing' (in radians: a value of zero implies that the pursuer is exactly behind); pursuer forwards velocity ($\mathrm{m \cdot s^{-1}}$); pursuer angular velocity (radians$\cdot \mathrm{s}^{-1}$); evader forwards velocity; evader angular velocity; pursuer energy (arbitrary units); and evader energy. The next two graphs are output activities of the pursuer's two 'neural' units, followed by six graphs showing outputs of the units in the evader's network (the networks are shown in Figure 10).

These two figures give some indication of the relatively rich behavior that can be produced by randomly connected noisy continuous-time recurrent neural networks. However, the behaviors exhibited by both the pursuer and the evader are somewhat distant from what we desire: the pursuer is supposed to chase the evader, but instead it starts by moving *away* from the evader. Similarly, the evader is supposed to be running away from the pursuer, but it's not: it moves towards the pursuer, and then (around t=4.3s) it starts turning in a tight circles for about five seconds. By this time, the pursuer has (as a result of random activity in its units) turned around and is heading toward the evader. It continues on a straight path, while the evader wanders around some more. The trial ends after the allotted 15 seconds without a collision: notice also that both animats have significant amounts of energy remaining at the end of the trial. Clearly, there is room for improvement here. But even though these two animats are bad, they are not as bad as others in their populations. Because this is the first generation, and the genotypes for the two populations were randomly generated by the same processes, at this stage some predators may run away at top speed from the evader; some evaders may actively seek the predator; and many animats in both populations will simply do nothing at all.

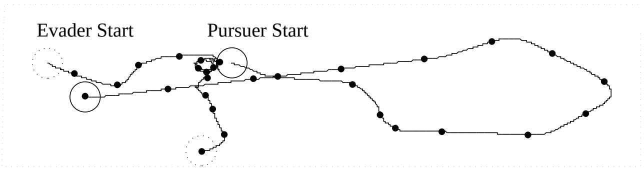

Figure 2: Pursuit-Evasion trajectory from generation 0. Plot shows start position of Pursuer (solid) and Evader (dotted): the circles are the same diameter as the animat 'bodies'. The paths taken by the two animats are marked with dots at one-second intervals, and the positions at the end of the trial are also shown (after 15 seconds). See text for further discussion.

Figure 4 shows the paths from a trial between the elite pursuer and evader after 200 generations of co-evolution. The pursuer is clearly chasing the evader, and the evader

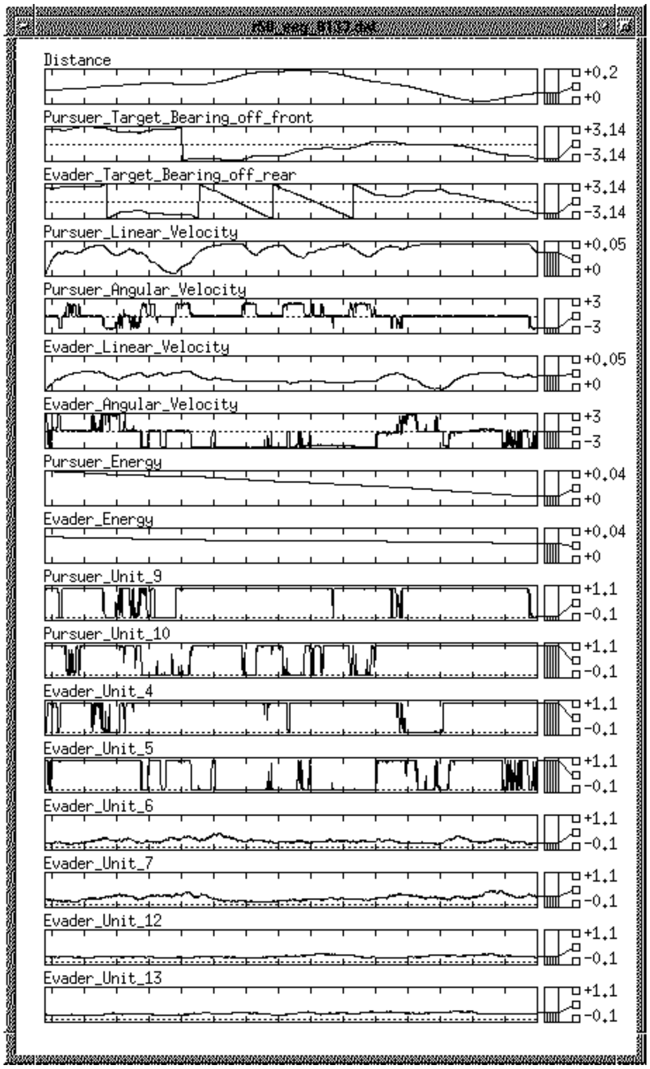

Figure 3: Data traces from the trial shown in Figure 2: horizontal axis is time, with tick-marks at one-second intervals. See text for explanation and discussion.

is clearly running away from the pursuer. But, as is illustrated in Figure 5, the pursuer strategy involves accelerating to near-maximum velocity, slowing only as a consequence of the need to slow one wheel for turning through differential steering. This policy consumes energy rapidly, and at around t=8s the pursuer's energy falls to zero: its motor outputs shut off, and it coasts to a halt. The evader has not been moving so fast, and hence has sufficient energy to last until very near the end of the trial, continuing to move away from the stalled pursuer. So, the pursuer is exhibiting the right kind of behavior but lacks subtlety, and the evader is also showing roughly appropriate behavior, although it is difficult to judge how it would fare against a more sophisticated pursuer.

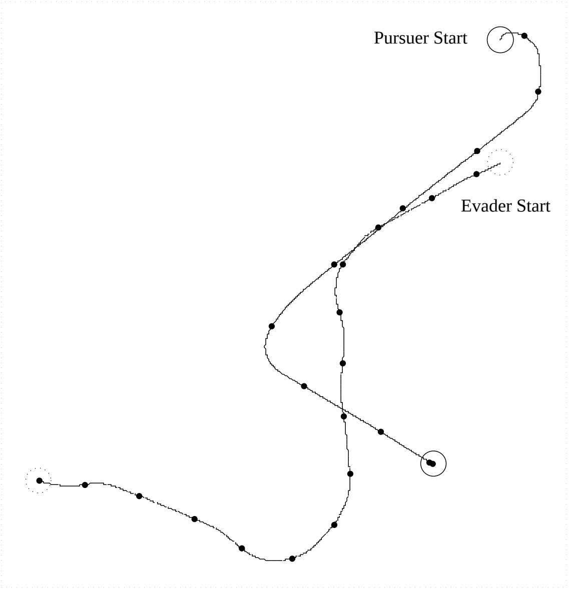

Figure 4: Pursuit-Evasion trajectory from generation 200. Display format as for Figure 2.

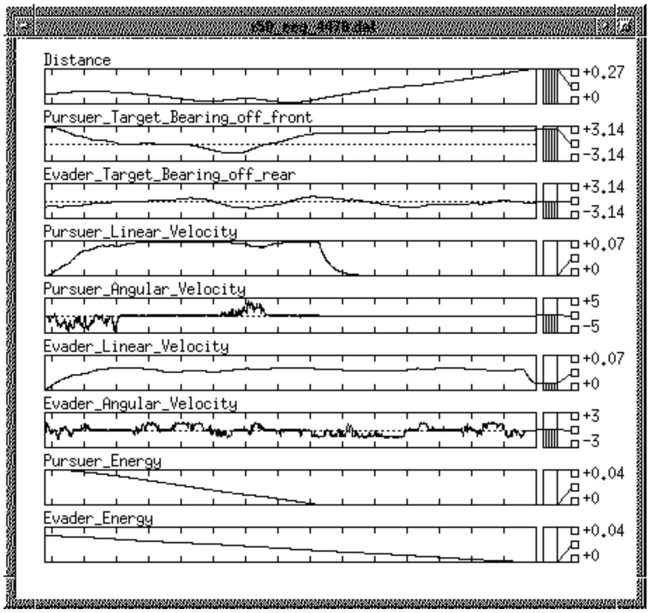

Figure 5: Data traces from the trial shown in Figure 4. Note that the vertical-axis ranges for distance, linear velocity, and angular velocity, have all been increased from those used in Figure 3.

Results from a contest between the elites of the two populations after 999 generations are shown in Figures 6 and 7. As can be seen both from the trajectory and from the graphs of target-bearings for both the pursuer and the evader, the pursuer keeps itself pointed straight at the evader, and the evader keeps the pursuer right on its "6-o'clock": again the pursuer uses all its energy and drifts to a halt, but its chasing strategy has forced the evader to use much more energy, and only about half a second later the evader also stalls. Presumably, slower-moving evaders have been selected against because they are more readily caught by the pursuers. This is not to say that a slower evader would *necessarily* be worse: a point we illustrate below. Nevertheless, at this stage in the coevolutionary process, neural network controllers for pursuit and evasion have manifestly been successfully co-evolved.

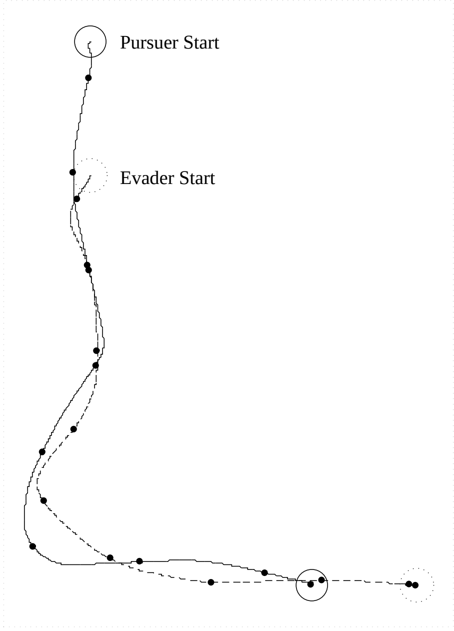

Figure 6: Pursuit-Evasion trajectory from generation 999. Display format as for Figure 2.

So, even after 999 generations, we have a pursuer which fails to catch the evader: this was also the situation at generation 200, and indeed at generation 0. As we discussed at length in an earlier paper [6], to check for progress or improvements in performance, it can be informative to test a current individual against ancestors of its current opponent(s). As an example, Figures 8 and 9 show trajectories and data traces from a trial where the best pursuer from generation 999 is pitted against the best evader from generation 200: as can be seen, the pursuer is indeed capable of catching some types of evaders, but the evaders at its generation are well enough adapted to its pursuit strategy that they get away successfully. It is notable that the generation-999 pursuer is markedly less accurate in chasing the generation-200 evader: it overshoots and corrects a couple of times before the final collision. This may be a sign that the ancestral evader is exhibiting a strategy which the pursuer is *less* well adapted to than the strategy of the evaders at its own generation: the generation-200 evader holds its speed at approximately 75% of the speed that the generation-999 evader runs at; the generation-999 pursuer may coordinate its turns and lunges with an 'expectation' that the evader is moving at the higher speed, and so it has some trouble with a slow evader from an ancestral generation: this is an example at the behavioral level of the effects of limited "genetic memory" [6].

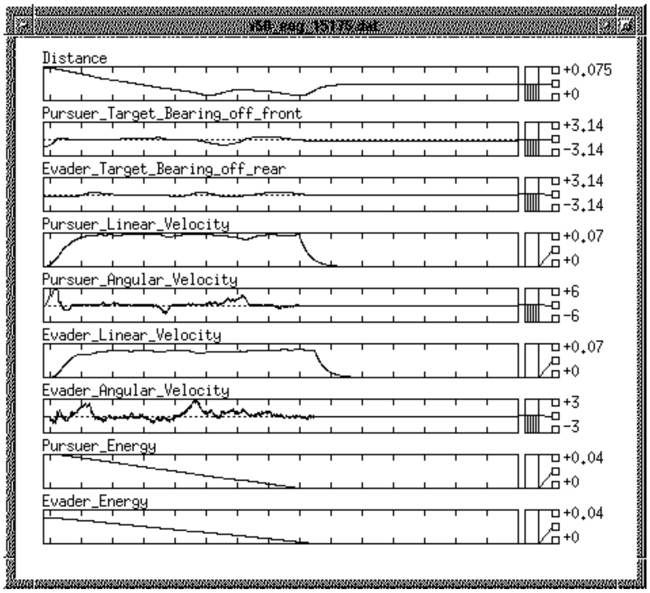

Figure 7: Data traces from the trial shown in Figure 6. Note that, in comparison to Figure 5, the vertical axis range for distance has been decreased, and the axis range for pursuer angular velocity have been increased.

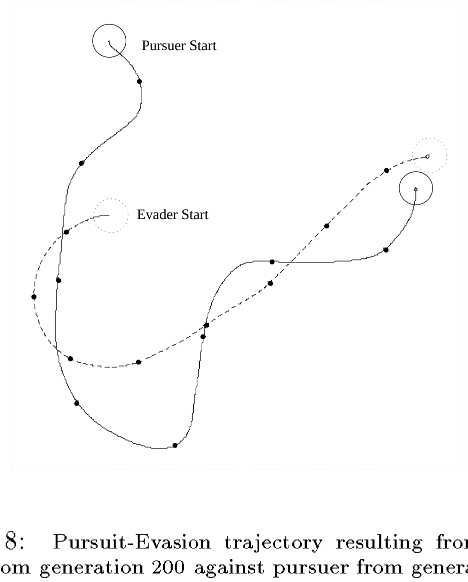

Figure 8: Pursuit-Evasion trajectory resulting from testing evader from generation 200 against pursuer from generation 999. Display format as for Figure 2. Trial ends with collision at t=8.6sec.

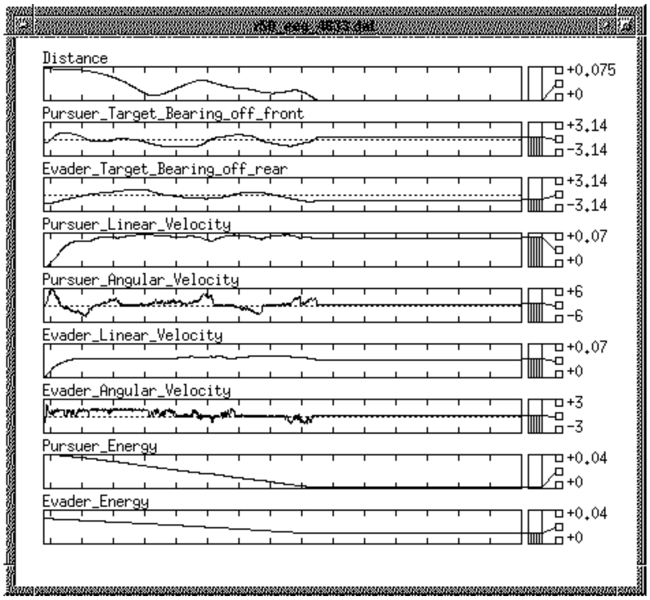

Figure 9: Data traces from the trial shown in Figure 8. Data recording continues after the collision at t=8.6sec, but all values are frozen.

The evolutionary changes in the morphology of the pursuer and sensory-motor networks are illustrated in Figure 10.

4 Further Work

Our work to date has demonstrated that the co-evolution of continuous-time recurrent neural-network controllers for pursuit and evasion strategies in environments with realistic sensory-motor dynamics is a reasonable thing to do. In this section we discuss some of the future directions this research could be taken in.

4.1 Applications in Biology

As we discussed in [10], there is a wealth of biological literature on pursuit and evasion in animals, with analysis at a variety of levels ranging from evolutionary arguments down to neuroethological studies of neural circuits governing pursuit/attack and evasion/escape.

In [5] it is noted that, just as artificial evolution can be used to semi-automatically propose new designs for the sensory-motor control architectures of artificial autonomous agents, so it could (in principle at least) be used to propose new 'designs' for neural sensory-motor coordination mechanisms in animals: the designs would effectively be models for the neuroarchitecture underlying an observable behavior in an animal. For this to be workable, it would be necessary to set up a simulation system where the physical and neural dynamics of the evolving animats are biologically plausible, and the evolutionary process only generates biologically plausible architectures (or selects very heavily against any implausible architectures): in selecting for animats which exhibit a particular behavior, the evolutionary process could generate a model or models for the neural mechanisms underlying that behavior, which may subsequently drive empirical or theoretical biology: an early example

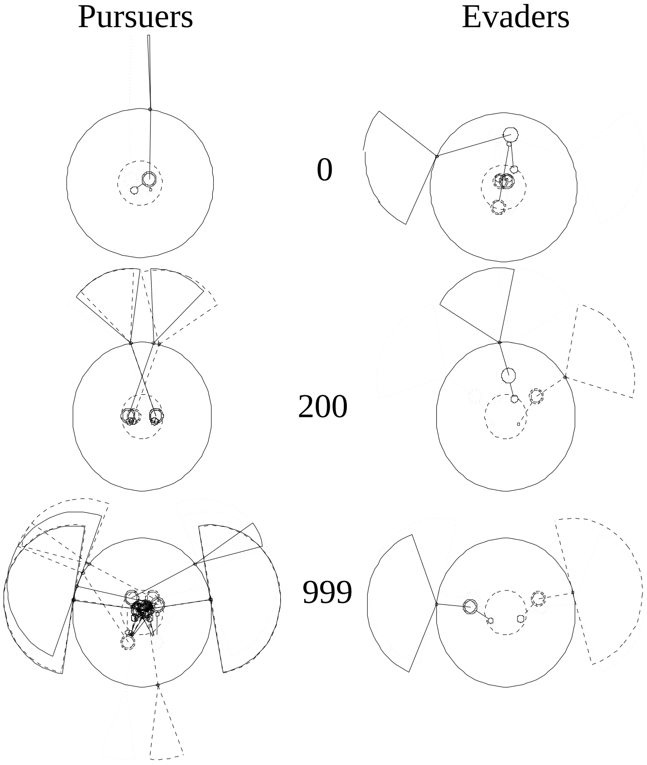

Figure 10: Plots of the sensory-motor morphologies of the elite pursuer and evader at generations 0, 200 and 999. Each plot is a top-down view of the animat, with the front toward the top of the page. The large circle is the extent of the animat's body. The inner dashed concentric circle marks a "spinal chord" zone on the body where the termination of a 'neuron' processing unit's output signifies that the unit contributes to the motor outputs of the animat. Other circles within the animat's body are 'neurons': the larger ones are neuron 'cell bodies' and the smaller ones are 'synapses'; the straight lines are connections joining neurons, via synapses, to each other or to motors or sensors. Visual sensors have genetically specified positions on the periphery of the animat's body, and each has a genetically-specified orientation (i.e. direction of view relative to the body) and angle of acceptance which are indicated here by the circle-segments emanating from the body perimeter. Different line styles are used to help distinguish different units. At generation 0, both the pursuer and the evader have Braitenberg-Vehicle-style architectures, with only two visual sensors each. After 200 generations, the combined effects of duplication and mutation can be seen: it is especially clear that the pursuer's six sensor units are close copies of each other; only a few small mutations have occured since the duplication events. By 1000 generations, the pursuer has a fairly elaborate sensory morphology, with four lateral wide-field photosensors, four frontal narrow-field sensors, and 2 rear-facing narrow-field sensors. In contrast, the evader has four laterally positioned wide-field sensors, without the frontal narrow-field or rear-facing sensors that the pursuer has.

of such work is [15]. Hence, one possibility is that we use our simulator for studying visually guided pursuit behavior in animals. Such behavior has been studied extensively in the biology literature. One system particularly well understood at the neural level is chasing behavior in house-flies (see e.g. [8]). If we attempt to evolve pursuer animats with fly-like dynamics, which produce fly-like behaviors, we may learn more about real flies.

Before such work can commence, it is likely to be necessary to make the (visual) environment more challenging. Currently, the only object in either animats' sensory world is another animat – i.e. its opponent. This abstracts away much of the difficulty of real visually-guided pursuit and evasion. In real worlds, it can be necessary to distinguish a moving target from a cluttered background (the image of which may also shift due to self-motion), or to deal with predicting the future path of a target to adaptively cope with brief occlusions. The fact that most of our evolved animats suffice with a small number of photoreceptors is probably due to the simplicity of the visual environment, and to evolve visual architectures of interest to biologists, it will probably be necessary to perform experiments in more visually complex worlds.

The most simple means of increasing visual complexity is to add static obstacles into the animat-world. Of more long-term interest (and complexity) is the possibility of having more than two agents in the environment at any one time. Even with only three animats, there are possibilities of exploring the effects of the system's (predetermined) dominance hierarchies: they could be defined transitively (giving a "food-chain" type scenario) or intransitively, forming a cycle; and it becomes possible to study issues in attention. As the number of animats in a trial increases, so it may be possible to study group behaviors such as pack-hunting and coordinated evasion (e.g. random scatter). Such group dynamics could also allow for studying signaling and communication [16].

Furthermore, in many species of prey (i.e. evader) animals, there are often three distinct phases to an encounter with a predator (i.e. pursuer) animal. At first the prey will be vigilant, conserving energy. When the predator comes closer, the prey will engage in linear fleeing, where it moves in a straight line at high speed, away from the predator. Once the predator is within some nearby distance threshold, the prey will then switch to the third phase, involving protean jinking and dodging behavior. Given the need for energy economy in our simulations, and game-theoretic arguments for the optimality of mixed (i.e. random) strategies in a variety of pursuit-evasion scenarios, it would be intriguing to see whether a similar three-phase evader strategy evolves without explicit selection for such phasic strategies. To keep the analysis clear, it would be necessary to work with non-differential-steer animats, so that forwards speed is not necessarily affected by making turns. We would then hope to see a negative correlation between pursuer-distance and evader speed (i.e. a primitive form of vigilance), and a nonlinear relationship between absolute angular speed and pursuer-distance, with a peak near to the threshold distance where dodging is initiated, and a range of distances for which the angular speed is approximately zero, corresponding to the distance range in which linear fleeing is exhibited.

Finally, we note that deciding whether a given behavior pattern can reasonably be referred to as adaptively unpredictable requires an operational definition of protean behavior, preferably one that can be expressed quantitatively. As yet, we have been unable to find an agreed-upon rigorous quantitative definition of protean behavior in the literature, which makes detecting proteanism in our simulations all the more difficult.

4.2 Applications in Entertainment

In the short-to-middle-term future, we believe that the market with the biggest potential for commercial application of our techniques is leisure and entertainment software (rather than, e.g., autonomous mobile robots). Co-evolutionary techniques are already being used to develop sensory-motor controllers and morphologies for virtual/software agents whose behavior makes them fun to watch or interact with, or makes them easy to choreograph in producing movie sequences involving groups of complex interacting agents (e.g. [13, 14]).

To indicate the possibilities, we have produced 3D movie sequences of the trajectories shown in Figures 2, 4, 6, and 8. The movies were made by taking the trajectory data shown earlier and using this to move two virtual 3D agents around a 2D plane: the animats are given 3D 'bodies' and 'eyes', purely for visual effect: Figure 11 shows one frame from one of the movies. A 'virtual camera operator' tracks the animats by moving the 'camera' in 3D as the chase unfolds, according to some simple rules. Of course, whether the movie sequences are entertaining is a matter of subjective opinion. But we think it is fair to say that they are a *much* more stimulating way of presenting the data than the 2D trajectory plots given here. Readers with access to the world-wide-web can find the movies (encoded as MPEG sequences) at the URL: `http://www.cogs.susx.ac.uk/users/davec/pe.html`.

4.3 Monitoring Techniques

Whether we are interested in co-evolution of pursuit and evasion for scientific purposes or for reasons of making fun software, there are still unresolved issues in monitoring and characterizing co-evolutionary dynamics. The indications are that experiments in co-evolution of artificial autonomous agents are, in general, likely to be highly resource-intensive (i.e take a lot of time and/or a lot of computing power: a run of 1000 generations typi-

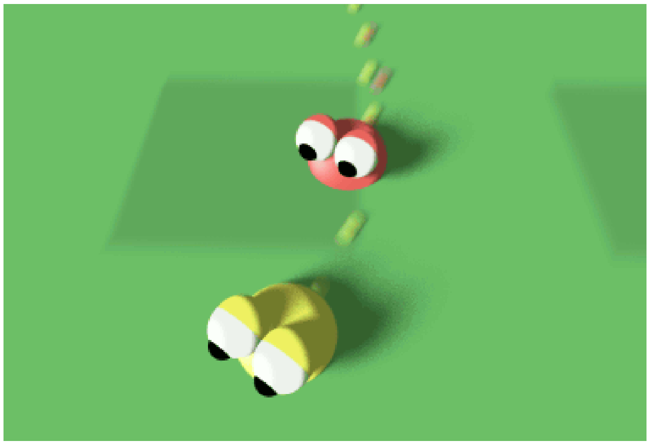

Figure 11: A frame from a computer-graphics pursuit-evasion 'entertainment' movie sequence: generation 999, same trajectory as Figure 6, 4.1sec into the trial.

cally takes about 3 weeks on an unladen Sun Sparc 20). As increasing magnitudes of resources are committed to such experiments, so there will be an increasing requirement for robust and informative monitoring techniques. The primary pragmatic need will be for techniques which make it possible to determine whether the evolutionary process is heading in the desired direction, or any direction at all. We discussed this problem and some tentative solutions in [6]. Our work in this area continues.

5 Conclusions

This paper has described our methods and presented tiny cross-sections of data from the torrent of megabytes that a complex co-evolutionary simulation can generate. But most of what we've learned from this project can't be expressed in numbers. We started out with several beliefs that turned out to be naive, simplistic, or simply mistaken; these included the conceits that (1) smart pursuers and evaders with ingenious nervous systems will evolve quickly under coevolution; (2) protean (adaptively unpredictable) behavior will emerge quickly, robustly, and permanently in almost any pursuit-evasion simulation; (3) almost any robot physics will lead to interesting pursuit-evasion strategies; (4) co-evolutionary progress is easy to measure; and (5) the details of evaluation functions, trial initialization methods, network parameters, morphogenesis methods, and genetic algorithm parameters won't matter much because coevolution is so robust. We know better now, and hope that other researchers will remain inspired by the grace and drama of real animals doing pursuit and evasion, but that they will be as cautious as we have become about the ability of computer simulation to capture the evolutionary dynamics that produced such behavior.

Acknowledgements

DC's work was supported in part by a grant from the UK EPSRC. GFM's work was supported by NSF-NATO Post-Doctoral Fellowship RCD-9255323, NSF Research Grant INT-9203229, the University of Sussex, the Univerity of Nottingham, and the Max Planck Society.

References

[1] R. D. Beer. On the dynamics of small continuous-time recurrent neural networks. *Adaptive Behavior*, 3(4):471–511, 1995.

[2] V. Braitenberg. *Vehicles: Experiments in Synthetic Psychology.* MIT Press Bradford Books, Cambridge MA, 1984.

[3] D. Cliff. NCAGE: Network control architecture genetic encoding. Technical Report CSRP325, University of Sussex School of Cognitive and Computing Sciences, 1994.

[4] D. Cliff, I. Harvey, and P. Husbands. Explorations in evolutionary robotics. *Adaptive Behavior*, 2(1):71–108, 1993.

[5] D. Cliff, I. Harvey, and P. Husbands. Artificial evolution of visual control systems for robots. In M. Srinivisan and S. Venkatesh, editors, *From Living Eyes to Seeing Machines.* Oxford University Press, 1996. In Press.

[6] D. Cliff and G. F. Miller. Tracking the Red Queen: Measurements of adaptive progress in co-evolutionary simulations. In F. Morán, A. Moreno, J. J. Merelo, and P.Chacón, editors, *Advances in Artificial Life: Proc. Third Euro. Conf. Artificial Life*, pp.200–218. Springer-Verlag, 1995.

[7] J. J. Grefenstette. The evolution of strategies for multiagent behaviors. *Adaptive Behavior*, 1(1):65–89, 1992.

[8] M. F. Land and T. S. Collett. Chasing behaviour of houseflies (*Fannia canicularis*). *Journal of Comparative Physiology*, 89:331–357, 1974.

[9] G. F. Miller. Protean primates: The evolution of adaptive unpredictability in competition and courtship. In A. Whiten and R. W. Byrne, editors, *Machiavellian Intelligence II.* Oxford University Press, 1996. In Press.

[10] G. F. Miller and D. Cliff. Protean behavior in dynamic games: Arguments for the co-evolution of pursuit-evasion tactics. In D. Cliff, P. Husbands, J.-A. Meyer, and S. Wilson, editors, *From Animals to Animats 3: Proc. Third Int. Conf. Sim. Adapt. Behav. (SAB94)*, pp.411–420. MIT Press, 1994.

[11] G. F. Miller and J. J. Freyd. Dynamic mental representations of animate motion: The interplay among evolutionary, cognitive, and behavioral dynamics. Technical Report CSRP 290, University of Sussex School of Cognitive and Computing Sciences, 1993.

[12] M. A. Potter and K. A. De Jong. A cooperative coevolutionary approach to function optimization. In Y. Davidor and H.-P. Schwefel, editors, *Proc. Third Conference on Parallel Problem Solving from Nature*, pp.249–257. Springer-Verlag, 1994.

[13] C. Reynolds. Competition, coevolution, and the game of tag. In R. Brooks and P. Maes, editors, *Artificial Life IV*, pp. 59–69. MIT Press 1994.

[14] K. Sims. Evolving 3D morphology and behavior by competition. In R. Brooks and P. Maes, editors, *Artificial Life IV*, pp. 28–39. MIT Press, 1994.

[15] D. Stork, B. Jackson, and S. Walker. 'Non-Optimality' via pre-adaptation in simple neural systems. In C. Langton, C. Taylor, J. D. Farmer, and S. Rasmussen, editors, *Artificial Life II*, pp.409–429. Addison Wesley, 1992.

[16] G. M. Werner and M. G. Dyer. Evolution of herding behaviours in artificial animals. In J.-A. Meyer, H. Roitblat, and S. Wilson, editors, *Proc. Second Int. Conf. Sim. Adapt. Behav. (SAB92)*, pp.393–399. MIT Press, 1993.