University of Southampton

Faculty of Engineering, Science and Mathematics

School of Electronics and Computer Science

Communication without dedicated signalling channels: A general finding?

By

Jordi Arranz

October 1st, 2010

A dissertation submitted in partial fulfilment of the degree of

PhD Complex Systems Simulation

By examination and dissertation

Consciousness generally has only been developed under the pressure of the necessity of communication.

Friedrich Nietzsche, 1882

After all, one way of casting this whole question (the way that I usually think about it) is not "How do we get from the bricks, amoebas, and then apes to us?" but "How in the world could you ever make a conscious automaton, how could you make a conscious robot?" The answer, I think is not to be found in hypotheses about hardware particularly, but in software. What you want to do is design the software in such a way that the system has a certain set of concepts. If you manage to endow the system with the right sort of concepts, you create one of those logical spaces that Jaynes talks about.

Daniel C. Dennett, 1998

Abstract

The central finding of Quinn (2001) was that communication can evolve in an evolutionary robotics context without the use of dedicated signalling channels. Quinn simulated fairly realistic robotic agents controlled by neural networks and equipped with proximity sensors and wheels for locomotion. The agents were set a coordinated movement task (i.e., to move their combined centre of mass as far as possible in a limited time frame). Non-coordinated strategies do very poorly at this task, but coordination was not trivial to achieve, as the agents had no pre-given way of signalling to each other. Evolutionary runs revealed that coordinated overall behaviour could in fact emerge from a dance-like movement pattern that allowed the two agents to spontaneously establish "leader" and "follower" roles. Quinn's result is very exciting because it shows the potential for ALife models to look at the origin of communication from genuinely non-communicative contexts. Other models that look at the conditions for the stability of a signalling system over a pre-defined signalling channel can only really refer to the evolutionary maintenance of communication rather than its beginnings. Although other evolutionary robotics researchers have referenced Quinn's result, this has typically been in the context of interpreting some evolved behaviour in their own experiments. The importance of Quinn's result for cognitive theorists interested in the evolution of language and social intelligence was acknowledged by Kirby et al. (2002) but this side of the work has not been pursued. Our project involves asking whether Quinn's findings are general. In other words, we have successfully replicated Quinn's central result but without using the particular simulation framework that he employed. Most of the results of our experiment match Quinn's results and therefore it is suggested that the emergence of communication from non-communicative origins is likely to be a common evolutionary adaptation to niches that involve the coordination of cooperative behaviour.

1 – Introduction

The emergence of intelligence and mind in living systems is a key question involving multiple research areas (Bedau et al. 2000). It is a grand challenge with a long history that, even today, poses multiple unresolved questions: what is mind, how did it evolve, and if we succeeded in reproducing it in an artificial system how would we know? Well-established philosophical positions (Dennett 1997 and Chalmers 1991) and an increasing number of empirical studies support a tight connection between the development of mind and language. Under the assumption that the emergence of communication comes before the emergence of complex cognition, the problem of reproducing mind in an artificial system becomes much more tractable: in other words, if we can explain how and why agents are communicating with each other, the job of explaining their mental states becomes easier.

General arguments for the idea that consciousness and mind may have a socio-linguistic origin can be found in many different disciplines such as psychology, anthropology, etc. However, the first explicit approach was published by Julian Jaynes in his book “The breakdown of the bicameral mind” (1976). Jaynes’s main thesis, based on psycho-historical analysis and neurobiological studies, states that the modern mind is a product of a highly evolved linguistic ability that was itself produced by increasing levels of social complexity. This evolutionary stage was not achieved until around 2000 B.C. and therefore humans living before that time had no sense of self-awareness. In such a state, the cognitive functions took place in the two hemispheres of the brain but there was no connection between them. In the absence of such a connection, auditory hallucinations acted as the missing communication channel, with half of the brain acting as a speaker and the other half obeying. The right hemisphere of the brain stores experiences and memories that are transmitted to the left hemisphere via auditory hallucinations, perceived by the subject as external voices. This kind of thought persisted until language acquired enough complexity to manage the internal dialogue in a more sophisticated way: what Jaynes calls metaphorical language. The book was strongly criticized when published, even though Jaynes’s hypotheses were supported by scientific studies in many different fields. In recent years, Jaynes’s theory has gained acceptance by many scientists like Daniel C. Dennett (1996), William H. Calvin (Calvin et al. 2000), Merlin Donald (2001) and others. New advances in functional brain imaging techniques have shown that auditory hallucinations take place only in the right hemisphere of the brain which explains why the hallucinated voices would be perceived as alien (Olin 1999, Cavanna et al. 2007). The implications of the bicameral mind are extremely important and impact many different scientific areas. Among them, two are most important in this context. The first is that mind-consciousness and language seem to be two sides of the same coin, and therefore approaches based on the evolution of language are promising ways to study the emergence of mind and consciousness. The second important insight from Jaynes’s work is that language and mind evolved gradually, driven by the increasing complexity of the environment (particularly the social environment).

Thus it seems that studying the origins of language and communication may be fundamental to understanding the mind and its origins. In this context, computational models that mimic the origins of language can be very useful in order to model the evolution of mind without having to deal with the associated epistemological issues. From an evolutionary point of view, the origin of language is problematic because it contains a paradox: the existence of a signal makes no sense if nobody can understand it (Maynard-Smith 1997). No signal could reasonably exist without a response and no response could exist without a signal. This paradox can be solved if we assume that both primitive signals and responses were non-intentional. When an individual feels some kind of danger, it is likely to express anxiety with some sort of uncontrolled and meaningless behavior induced by the stress of such situations. (These expressions of anxiety need not be functional in any way; they can be merely epiphenomenal.) Other individuals may react to this signal but not because they associate it with any particular meaning; simply because the signal itself happens to induce a somatic reaction. Something similar happens when an arbitrary sound alerts or startles us, even if that sound has no natural meaning in our environment. Reproducing and mimicking these uncontrolled behaviors outside their original context is probably one of the earliest forms of intentional and conceptual communication. Since the existence of a dedicated communication channel such as speech makes no sense at this evolutionary stage, any implicit communication that does occur must take place through an existing mechanism, i.e., gestures and body movements.

Before 2001, all the existing computational models of the origins of language were designed with explicit communication channels. In 2001, Quinn showed that communication between agents could be evolved without specific communication channels. In Quinn's model, instead of using fully developmental representations of the individuals, only the brain, in the form of a neural network, was evolved. The model was based on a population of simulated Khepera robots equipped with local sensors (short range IR). The individuals of the population were evaluated in pairs, forcing them to cooperate in order to solve a particular co-ordination problem, so the fitness of each individual was highly coupled with the fitness of its paired partner. The simulation showed that basic communication evolved from functional but non-communicative behavior in form of dance-like movements. Another important result is the emergence of a hierarchy: one of the robots leads and the other follows despite none of them is intrinsically biased to adopt a particular role.

Despite Quinn's work not having been pursued by cognitive theorists it is very exciting because it shows the potential for ALife models to look at the real origin of communication, rather than just the conditions under which it could be maintained in a system where it was already possible. However, what if Quinn's result was a freak occurrence, and turned out to be related to some detail of the robot's sensory system or cognitive architecture? We think that precisely because Quinn's result is far-reaching, it is very important to establish its generality before going further.

2 – Simulation Model

Since the aim of this work is to compare and generalize Quinn's results, we have set up the same experiment in a completely different framework. Instead of using a Khepera robot simulator, a different hand-made 2D simulator was populated with pairs of a different type of agent. The agents employed are of the same size and shape as a Khepera robot but the sensors are of a different kind, number, and position. Since the new agents do not have to be realistic representations of a particular robot, motor-wheels are completely avoided and movement is just a translation and/or rotation in the simulated 2D world. Instead of using a continuous-time recurrent neural network to control the agents, as Quinn did, our model is built around a rule-based system with reinforcement learning (effectively a learning classifier system or LCS). In order to be able to compare both computational models the task the agents face should be exactly the same as in the original work: two agents must be capable of moving together for as great a distance as possible while staying within each other's sensor range and without colliding. While using the same kind of genetic algortihm (GA) to evolve the population of agents, the parameters employed, as well as the way fitness is computed, are different. As in the original model, there are no dedicated communication channels; nor are there predefined roles or commands. Agents are evaluated in pairs and given a certain amount of time to solve the task. Evaluation is performed in discrete time steps, at every time step new sensor values are computed for both agents, and finally the agent behaves according to its sensory input through the application of a rule. Every pair of agents gets the same score depending on their performance while solving the problem. A selection process keeps the best agents and deletes the worst of every generation, with new agents being created through recombination and mutation of the successful individuals of the previous generation.

Our strategy here is to partially replicate Quinn's paradigm and to see whether the same results emerge. We believe that the differences between our computational model and Quinn's are significant enough that succesful replication of the results would do much to establish the generality of Quinn's central finding.

2.1- Agent

Quinn's agents are fairly realistic simulations of a Khepera robot. Khepera are cylindrical-shaped agents with two independent motor-wheels that provide movement and rotation capacity. A set of eight IR proximity sensors give the robot the ability to perceive nearby objects (fig.1).

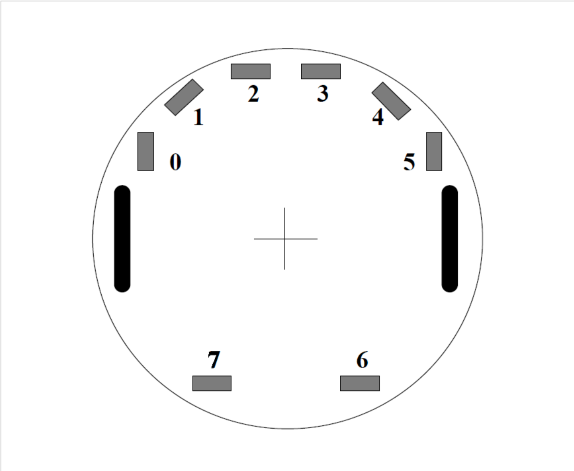

Figure 1. Khepera robot with eight IR sensors and 2 motor-wheels (Quinn, 2001).

The original agent has been simplified in several ways while maintaining enough similarities to be able to take on the coordination task (fig.2).

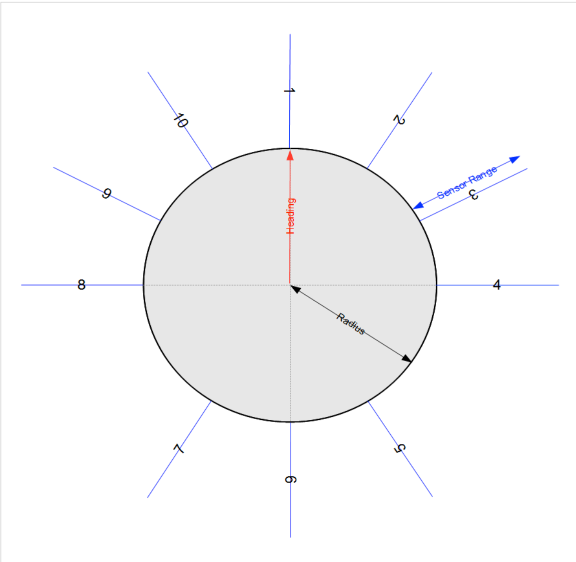

Figure 2. New agent with 10 ray-cast sensors.

The cylindrical shape of the robot has been kept in order to make the agents rotationally invariant and thus avoid any simple mechanism for detecting the orientation of one agent by its partner. Motor-wheels are no longer simulated since the agent is not a realistic simulation of a real robot. Movement and rotation are just transforms on a two dimensional space; agents are moved and rotated around their center-point. The eight IR sensors have been replaced by ten ray-cast sensors. Unlike IR sensors, ray sensors have a narrower sensing area. They operate by throwing a ray of a certain length and infinitesimal width along the vector the sensor is pointing to. If the ray collides with some object the sensor reports the collision distance, otherwise the sensor reports 0. Since the amount of space these sensors are able to scan is significantly smaller than IR sensors, two additional sensors have been added, bringing the total to ten per agent.

2.2 – Learning Classifier System

Initially developed by Holland (1976), Learning Classifier Systems (LCSs) are adaptive rule-based systems based on genetic algorithms (GAs) and reinforcement learning (rule scoring). While classical rule-based systems operate using a static rule set, LCSs are able to discover new rules (adaptive) with the aid of a GA. Rules are scored in some way such that the GA can evaluate the worth of existing rules and generate promising new ones. The LCS employed does not correspond exactly with any of the main types described in the literature but it keeps the general idea of evolving a set of rules and using a scoring method in order to fire the most promising rules when more than one is matched.

ID	S1	Lop 1	S2 2	Lop 2	S3 3	Lop 3	S4 4	Lop4	S5	Lop 5	S6	Lop 6	S7	Lop 7	S8	Lop 8	S9	Lop 9	S10	Aop	Effector
0	#	NOR	#	NAND	46297	OR	26897	AND	#	XOR	21026	XOR	47552	NOR	47547	OR	#	NAND	46465	GTR	FRWD

Figure 3. Example classifier with ten sensors, nine logical operators, one arithmetic operator and one effector.

Every agent in the population is generated with ten random classifiers. A classifier contains a set of sensor values (one for every sensor), several logical operators, a single arithmetic operator, and a single effector (fig.3). When the sensory inputs of the agent match the sensor values of a classifier, the classifier is “fired” and its effector applied.

ID	Binary Logical Operators	Logical Expression
0	AND	A AND B
1	OR	A OR B
2	XOR	(A OR B) AND NOT(A AND B)
3	NOR	NOT(A OR B)
4	NAND	NOT(A AND B)

Figure 4. Logical operators detail.

In order to know which classifier matches an agent’s sensory input at a certain time step all of them should be checked. The evaluation of a classifier starts comparing every agent sensor with its corresponding value sensor in the classifier through the classifier’s arithmetic operator (fig.5). The result is a chain of logical values (true or false depending if the condition is met or not). The logical chain is evaluated from left to right with the classifier logical operators (e.g., TRUE AND FALSE NAND TRUE etc., see fig.4). If the final result is true, the classifier is added to a list of matched classifiers. When all the classifiers have been checked, the classifier with the highest score among the matched ones is fired.

ID	Binary Arithmetic Operators	Arithmetic Expression
0	EQ	EQUAL
1	LSS	LESS THAN
2	GTR	GREATER THAN

Figure 5. Arithmetic operators in detail.

Firing a classifier means the agent is going to execute its effector (fig.6). If the classifier being fired has a FORWARD effector, the agent will move forward a certain amount of space depending on its velocity and the size of the simulation time step.

ID	Effector	Result
0	NONE	DO NOTHING
1	FORWARD	MOVE FORWARD
2	BACKWARDS	MOVE BACKWARDS
3	ROT_CW	ROTATE CLOCKWISE
4	ROT_CCW	ROTATE COUNTER CLOCKWISE

Figure 6. Effectors in detail.

Classifiers are scored when fired depending on their immediate effect on the fitness of the agent at a specific time step (Eq.4). The effect of the action of a classifier in the fitness of an agent is just the difference between the current and previous time step fitness.

$$Classifier\ Fitness = \frac{\sum_{k=1}^{K} [Af_{current} - Af_{last}]}{K} \quad (Eq.5)$$

Equation 5. Fitness of a classifier on a timestep.

In some cases it is not possible to match the inputs with any of the classifiers. In such cases, a new random classifier is generated that matches the current sensor values. The new classifier is immediately added and fired.

2.3 – Genetic Algorithm

Efficiently exploring the classifier space is a key point if we want the agents to evolve interesting behaviours. A simple generational genetic algorithm has been used to evolve the populations. At the beginning of every run a new population of random individuals is generated. Different operators and effectors have the same probability to appear while the probability of a wildcard in a sensor is 0.1. At the end of each generation the 60% of individuals with the highest fitness scores are kept and the remaining 40% are deleted from the population. New individuals are created by recombination and mutation of the survivors. Recombination is performed by picking two random individuals among the survivors and generating a new one by uniformly combining the classifiers of both parents. Variation of the new individuals is added through mutation. Mutation is performed by shifting by one unit (like a circular array) the index of a sensor value, operator or effector (figs. 4, 5 and 6). Every component of the classifier (sensors, operators and effector) have an independent mutation probability of 0.01, therefore the mutation probability of the whole classifier is 0.21 (a classifier has 21 components).

Evaluation of agents is done in pairs; every agent starts within sensor range of its partner and with a certain distance and angle.

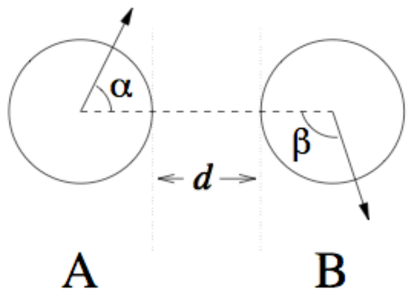

Figure 7. Initial distance and angles of a pair of agents (Quinn, 2001).

Every agent in the population is evaluated from a set of different initial distances and positions against many different partners (fig.7). The initial distances are *2, 2.5* and *3* cm while the initial angles are *pi/2*, *pi*, *3pi/2* and *2pi* rad. Since the population size is 25 and subtracting the rotational duplicates every agent is evaluated exactly 48 times (24 times by 2). As Quinn stated in the original work, the number of evaluations with different partners in different initial configurations have an important impact on early stages of evolution thus in the later discovery of interesting behaviours

The performance of a pair of agents at the end of an evaluation is shown in Eq.1. The fitness of every time step is normalized, added to the final distance parameter D, and averaged.

$$Pair\ Fitness = \frac{D + \frac{\sum_{t=0}^{S} [f(d_f)]}{S}}{2} \quad (Eq.1)$$

Equation 1. Fitness of a pair at the end of an evaluation.

The fitness of an agent on a certain time step (Eq.3) is computed as an exponential decay of the distance to the other agent. If an agent is in sensor range the fitness obtained is 1, otherwise fitness decreases exponentially with the distance.

$$D = \begin{cases} \frac{max[D_a, D_b]}{D_{Target}} & , D_a < D_{Target} \wedge D_b < D_{Target} \\ 1 & , D_a \geq D_{Target} \wedge D_b \geq D_{Target} \end{cases} \quad (Eq.2)$$

Equation 2. System distance parameter.

The maximum distance is computed as the maximum linear distance an agent can achieve regarding its linear velocity and simulation time (Eq.4).

$$f(d_f) = \frac{(1\text{e-}10)^{\frac{d_t}{d_{max}}}}{d_{max}} \quad (Eq.3)$$

Equation 3. Fitness of a pair in an evaluation timestep.

At the end of the evaluation, the final fitness is computed considering the displacement of both agents from their initial point (Eq.1). If both agents moved more than the target distance (25 cm.) the D parameter is 1, otherwise D is the quotient of the biggest distance and the target distance (Eq.2).

$$d_{max} = (2\text{TV}_{linear}) + d_0 \quad (Eq.4)$$

Equation 4. Maximum linear distance.

At the end of a generation the final fitness of an agent is the average of the different scores in each evaluation.
One of the important differences with the original work when computing the fitness is the absence of a collision term. In the original computational model, the number of collisions of a pair during its evaluation was used as a modifier of the final fitness (more collisions imply less fitness). In the first stages of development of our model we did make use of a collision factor, but we realized that it was not strictly necessary since the agents tended to avoid collisions as the evolutionary process went on. The reason for that is because when two agents are colliding they cannot “see” each other because they are out of range (negative distance or distance less than agent’s diameter) thus the fitness given is 0, and therefore colliding behaviours will tend to disappear at some point through selection, without the need for an explicit penalty term.

3- Analysis

Eleven runs have been performed in this experiment. The different parameters employed are shown in figure 8. The angular velocity was found to be critical since in most evaluations one of the agents should be able to rotate near 2pi radians in order to align its front left sensors, thus if angular velocity is too small compared to the simulation time, the agents will never evolve some of the basic behaviours needed to accomplish the task. Other important parameters like mutation rate and elite size have been tuned by trial and error until finding appropriate values. The parameters related to the LCS were adjusted taking into account performance issues. The rest of the parameters are equal or similar to those employed in the original work. Note that it is central to our strategy here to look at whether Quinn’s basic results will hold given a different context and thus differences between the two experimental setups are deliberate.

Population Size: 25
Generations: 2000
Elite Size: 60
Mutation Rate: 0.010
Initial Number of Classifiers: 10
Simulation Time: 15.00 sec.
Simulation Time Step: 0.25
Target Distance: 25.0 cm.
Sensor Range: 5.0 cm.
Angular Velocity: 0.79 rad/sec.
Linear Velocity: 8.00 cm/sec.
Agent Radius 6.0 cm.
Number of Sensors 10
Initial Distances: 8.00 6.50 7.00
Initial Angles: 1.5708 3.1416 4.7124 6.2832

Figure 8. Initial Parameters of the computational model, including agent physical parameters, GA parameters and simulation parameters.

At the beginning of every run a new population of 25 individuals is generated, and each individual is initialized with a set of 10 random classifiers. Populations are allowed to evolve over 2000 generations. Data recorded for each generation includes maximum and average fitness scores (figs.9 and 10), the controller of the elite agent, and the elite agent's firing pattern. The full history of the best pair in every generation is also recorded for further offline analysis in a 3D visualization tool designed for this task. Every generation took 2.63 seconds to complete. The whole experiment took around 20 hours on a quad core laptop running Windows 7. The memory footprint was kept very low consuming only around 15 Mbytes average RAM with the above simulation parameters.

Among the eleven runs, 3 of them evolved perfect agents with fitness 1 (27%) and 6 of them evolved agents with a fitness score equal or greater than 0.975 (54%). Figure 9 shows the average best fitness across the 11 runs and the standard error bars. Figure 10 shows the average fitness across the runs and the associated standard error bars.

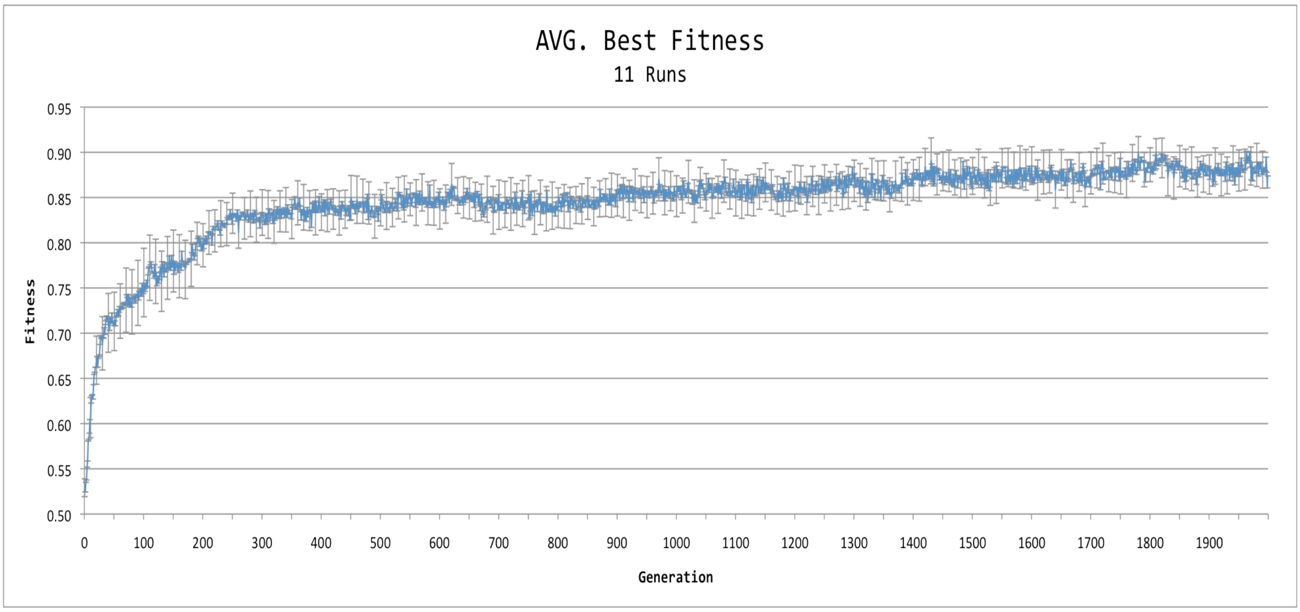

Figure 9. Average of the maximum generation fitness of 11 runs. Error bars show the standard error each 10 generations.

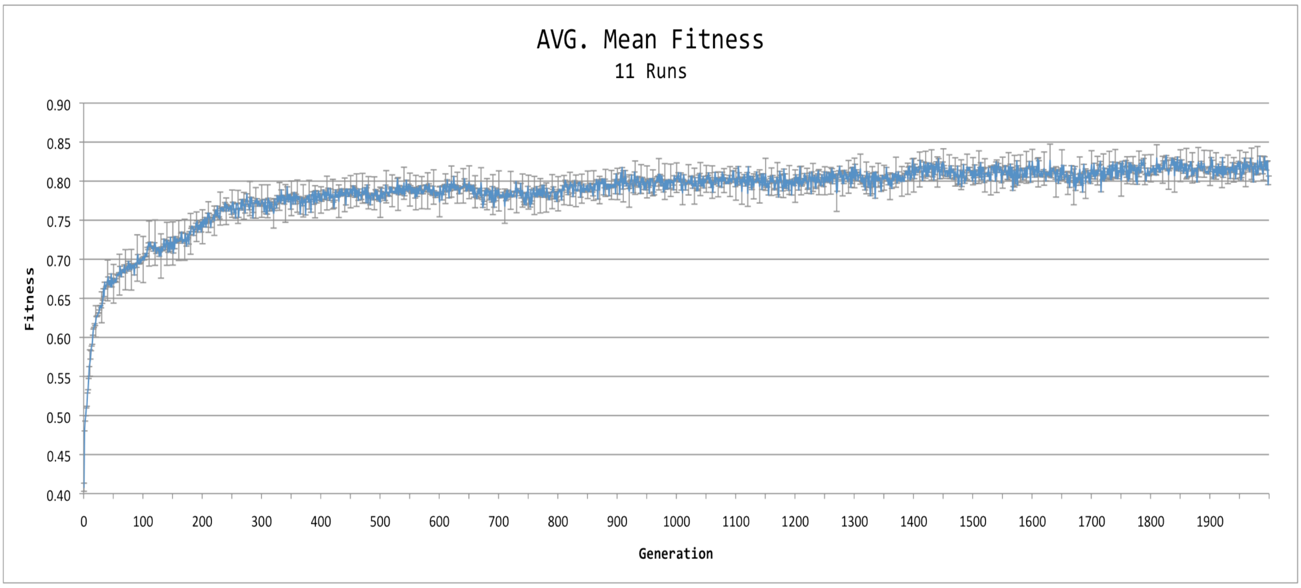

Figure 10. Average of the mean generation fitness of 11 runs. Error bars show the standard error each 10 generations.

3.1 – Evolved Behaviour

In order to achieve the coordination task, the successful agents evolved a set of different sequences of movements or behaviours. Behaviours are "fired" at very specific situations and not all of them are needed on every evaluation; depending on the various initial positions and angles, more or less sequences of movements are employed. The number of classifiers involved in each behaviour is variable: some of them need just 1 classifier and others need 2 or even 3 classifiers. Surprisingly the behaviours evolved by different successful agents on different runs show important similarities. There are 4 different behaviours: Detection or Alignment, Forward-Backwards (FB), Orbit and Bullet-like.

During the detection step, an agent rotates counter clockwise until its front right sensors detect its partner (fig.11).

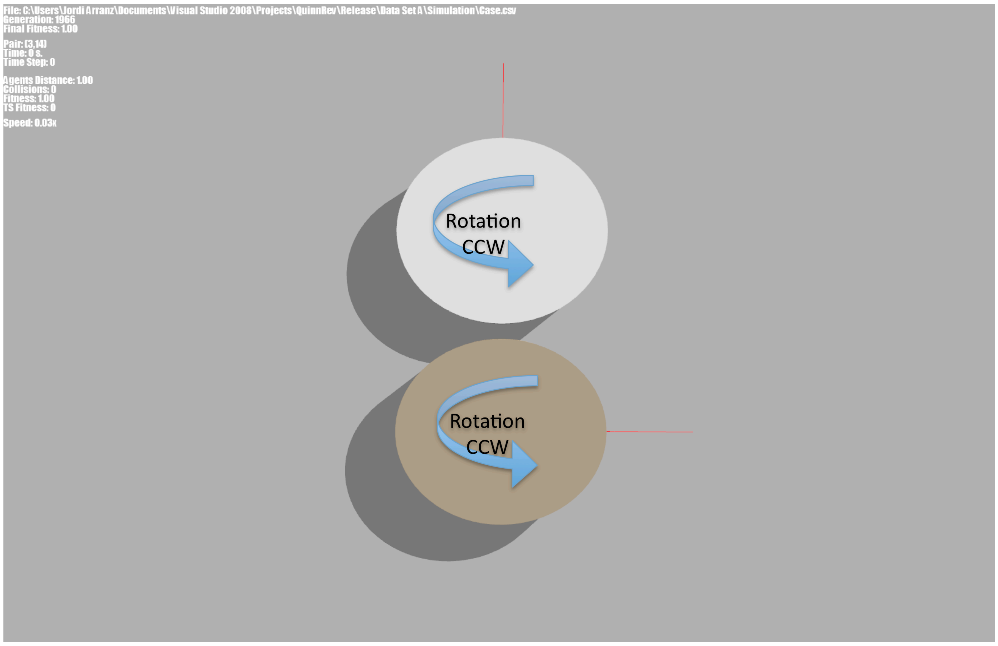

Figure 11. Simulation screenshot of the Detection-Alignment stage. Both agents will rotate counter clockwise until one of them gets aligned. Red lines show heading vectors. Note that the lower, brown agent will detect the upper, white agent first.

FB is a forward-backwards movement having two different functions, the first one is signalling to its partner its readiness and the alignment angle, the second one is “observing” if its partner has finished its own detection process by looking at the distance between them (fig.12).

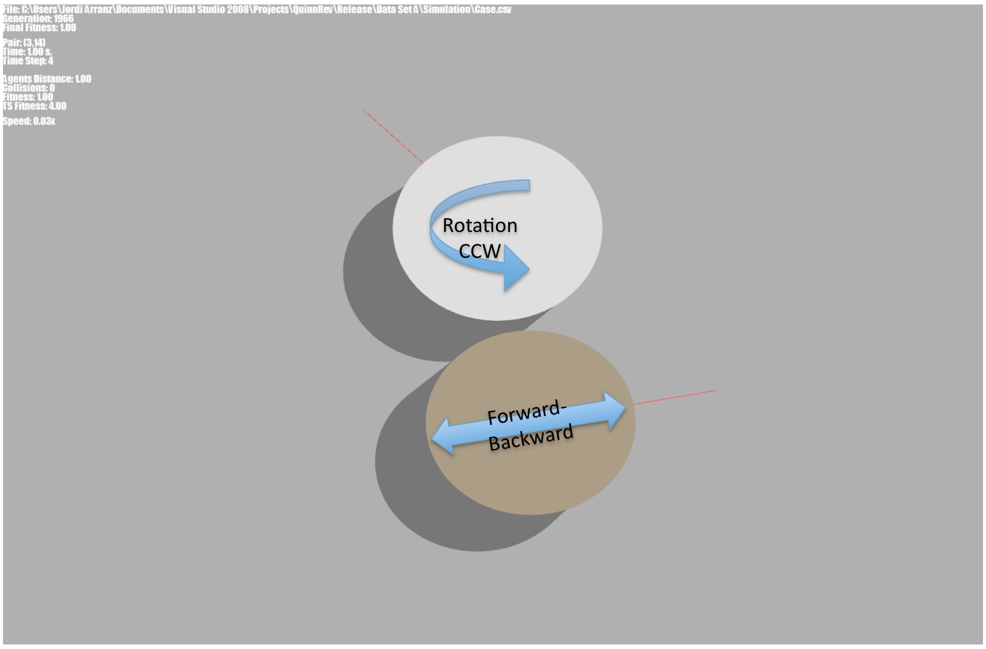

Figure 12. Simulation screenshot of the FB stage. The first aligned agent starts to signal its readiness moving forward and backward while the other agent is still rotating. Red lines show heading vectors.

The Orbit behaviour is the next in the sequence. Once an agent is aligned, it maintains its distance at a certain range, orbiting around its partner with a sequence of forward and rotation movements while their heading vectors remain in parallel (fig.13).

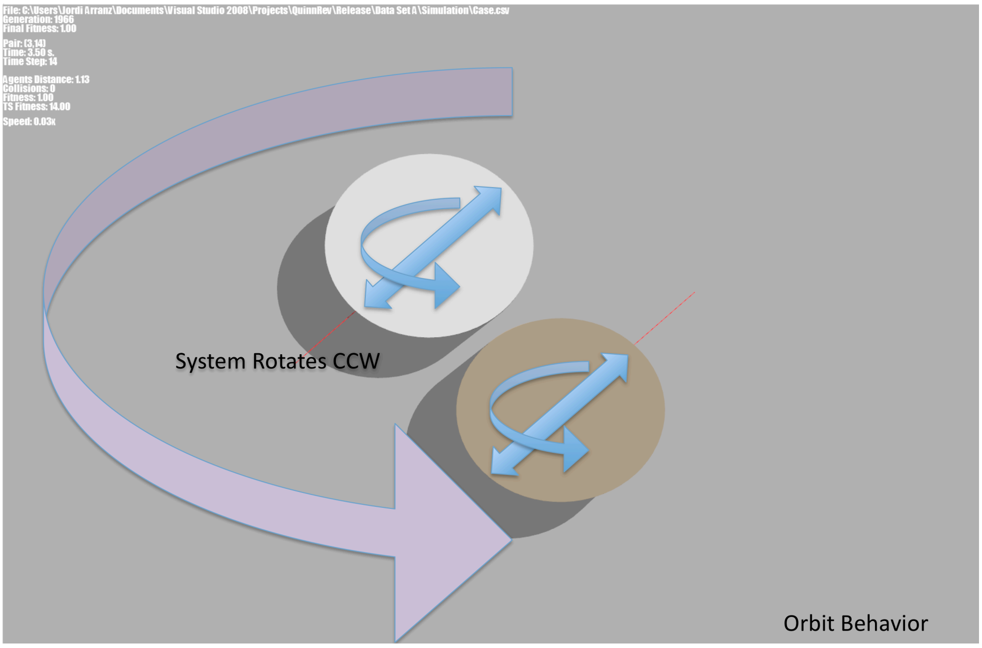

Figure 13. Simulation screenshot of the Orbit stage. Agents synchronize moving forward and backwards while rotating to maintain their distance. Heading vectors remain in parallel. Red lines show heading vectors.

While both agents are synchronized, the leading agent (first agent to align) starts going forward while the follower goes backwards (fig.14).

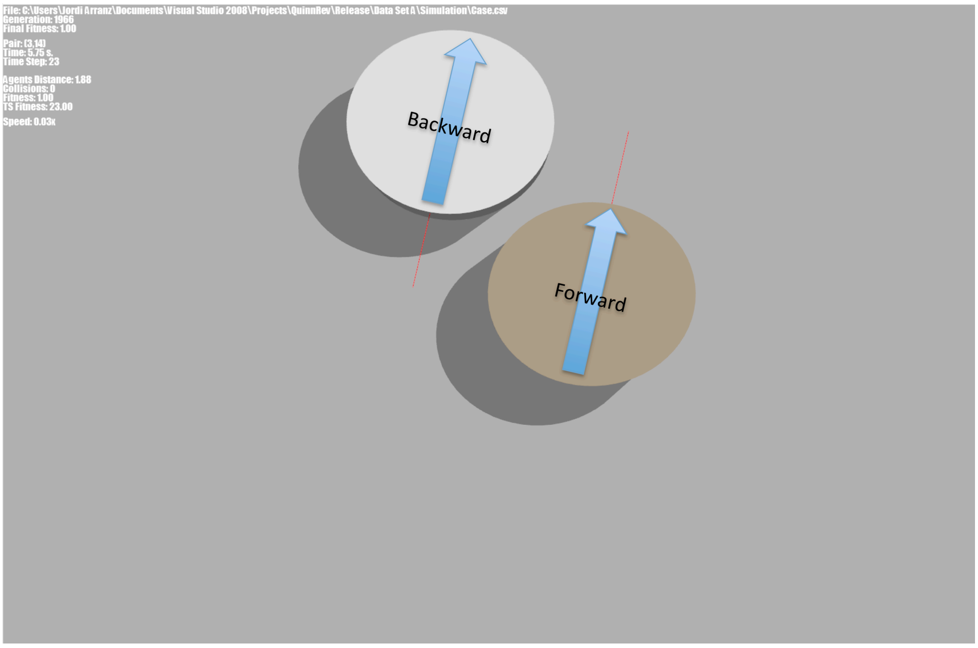

Figure 14. Simulation screenshot of the Bullet stage. The leader agent starts going forward while the follower goes backwards. Red lines show heading vectors.

Usually the whole process is as follows: both agents start rotating counter clockwise (fig.11) and when the first agent gets aligned it starts going one step forward and one step backwards while its partner is still rotating (fig.12). When agent B gets aligned too, both agents start the final synchronization process, orbiting around their joint centre of mass while maintaining a constant distance between themselves (fig.13). Finally, the first agent to achieve alignment goes forward and the second backwards until the end of the simulation (fig.14). It is interesting to note that, once both agents have finished their alignment, their relative orientations will not change anymore; after the alignment agents will rotate and move but in a synchronized way (their heading vectors remain in parallel). The case described is one of the typical cases but other cases exist in which there is a change of the leadership roles while agents perform the last synchronization step (Orbit).

One of the advantages when working with evolving rule-based systems (instead of neural networks for example) is the legibility of the evolved behaviours. In this case, most of the legibility is lost due the complexity of the classifiers. Therefore, the analysis has been done focusing on different parameters other than the classifiers, such as firing patterns. Since most evaluations are similar but not equal, we are going to take a look at a particular agent at a certain run (4), generation (1966) and evaluation (2) to accurately describe how a pair of agents behave in order to complete the task. In figure 15 we can see an evolved controller of an agent of fitness 1 evolved in run 4. The controller is composed of 13 classifiers which means that 3 classifiers have been added as a response to situations where no classifier could be matched. Among them, only 4 classifiers are functional, which means the other 9 classifiers are never used in this evaluation. Figure 13 shows which classifiers are fired at every simulation time step, with different colours being used to represent the different behaviours.

ID	S0	Lop1	S1	Lop2	S2	Lop3	S3	Lop4	S4	Lop5	S5	Lop6	S6	Lop7	S7	Lop8	S8	Lop9	S9	Aop	Effector
0	36542	OR	30537	AND	37397	AND	13214	NOR	1566	AND	29253	AND	10077	NAND	29462	AND	7721	NOR	5439	GTR	ROT_CW
1	#	NOR	41459	OR	49399	OR	28410	AND	33043	OR	13844	NOR	40253	XOR	23567	NAND	7016	OR	36286	EQ	BKWD
2	#	NOR	#	NOR	112	NAND	35477	AND	40011	AND	6096	AND	4878	NAND	6068	NAND	49186	XOR	15354	LSS	BKWD
3	10144	NAND	22067	NAND	15837	NOR	35223	NOR	1091	XOR	22480	XOR	12990	AND	36795	XOR	18469	AND	#	LSS	BKWD
4	40766	NAND	6139	AND	24015	NAND	1098	NOR	7740	NOR	44622	XOR	42282	NAND	26064	XOR	4996	OR	31607	GTR	BKWD
5	31574	OR	33118	OR	16074	AND	12397	NAND	6501	AND	27427	OR	772	OR	19199	XOR	3454	AND	178	GTR	FRWD
6	34001	OR	#	AND	48792	NAND	36758	XOR	21062	AND	46726	NAND	1605	NAND	4919	XOR	41151	NAND	29628	GTR	BKWD
7	35299	NAND	26363	OR	14379	XOR	#	NAND	23962	XOR	5197	OR	21192	NOR	28352	XOR	30168	AND	7571	LSS	FRWD
8	28640	NOR	24170	OR	23603	AND	44778	XOR	16129	NOR	12349	NAND	20769	NOR	47372	NAND	13318	AND	34696	EQ	FRWD
9	49881	XOR	20555	OR	1413	NAND	39807	NOR	27870	XOR	8881	AND	7214	NAND	49864	NAND	2349	NOR	10632	GTR	NONE
10	7758	NAND	49473	XOR	43881	XOR	9704	NAND	32819	NAND	47867	OR	44729	AND	44962	XOR	20523	NOR	30318	LSS	BKWD
11	16139	XOR	24035	OR	18876	NAND	18785	XOR	15574	AND	0	AND	43889	NAND	0	NAND	33973	OR	14646	LSS	BKWD
12	18779	AND	13429	AND	29302	XOR	0	NOR	18192	NAND	37749	NAND	45472	AND	14665	NAND	2550	NAND	17825	GTR	BKWD
13	0	NOR	38502	AND	32671	NAND	18412	AND	3670	NAND	28528	NAND	9032	XOR	30338	XOR	0	AND	20693	GTR	ROT_CCW

Figure 15. Evolved controller in run 4, classifiers in bold fired during evaluation 2 of generation 1966.

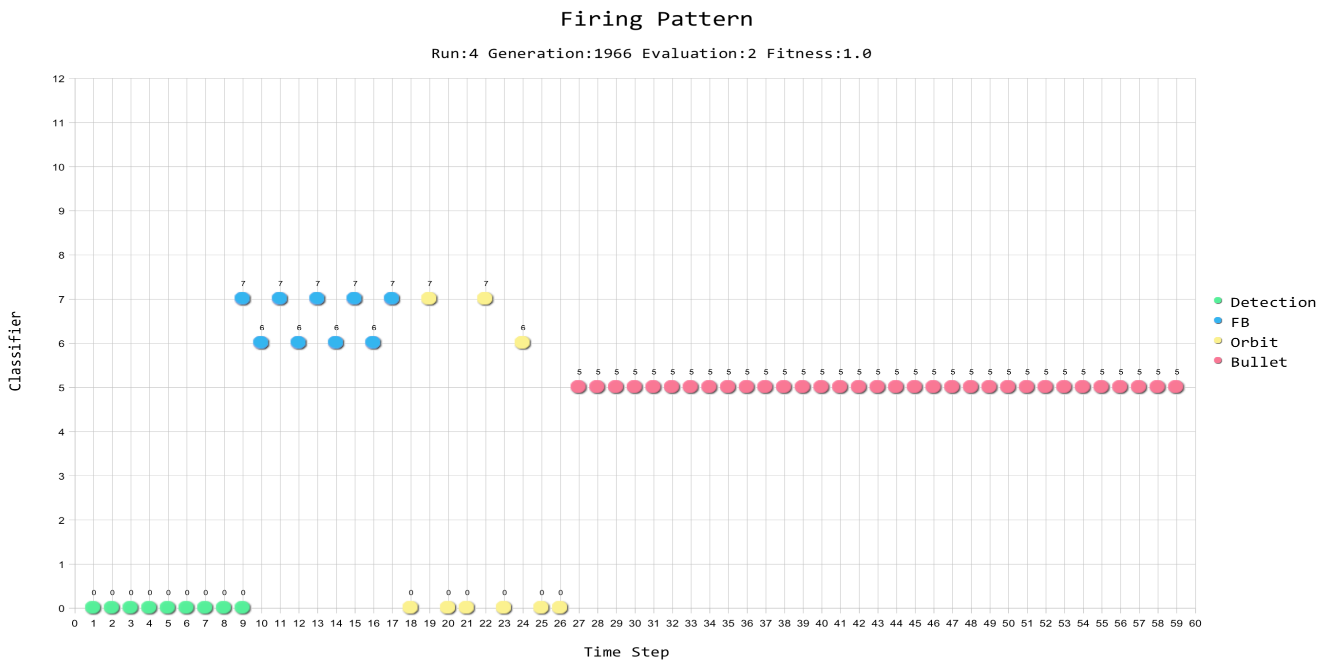

Figure 16. Firing pattern of an evolved controller in run 4, generation 1966, evaluation 2. Colours show the different behaviours employed by the agent in order to complete the task.

Evaluation starts with our agent not yet aligned; as a result it starts firing the Detection-Alignment classifier (see fig.16, green dots, and fig.15, classifier 0). That makes the agent rotate counter-clockwise until its front left sensors detect its partner. Once the agent is aligned, the FB behaviour becomes active until its partner gets aligned too (see fig. 16, blue dots, and fig.15, classifiers 6 and 7). Once both agents are aligned, their relative orientation will not change any more. In order to accomplish the final synchronization step, both agents have to move and rotate while keeping their relative orientation (see fig.16, yellow dots, and fig.15, classifiers 7, 6 and 0). Finally, when both agents are synchronized, the one that was aligned first (the leader) starts to go forward, and its partner backwards (the follower).

3.2 – Evolution of Behaviour

We have seen which behaviours have been evolved in order to accomplish the coordination task and their role in the different stages of the coordination. The appearance of these mechanisms takes place at very different points along the evolutionary timeline and they emerge as uncorrelated solutions to local problems whose solution increase the fitness of the pair. As evolution goes on, these solutions became more and more linked among themselves, eventually giving rise to the "global" solution.

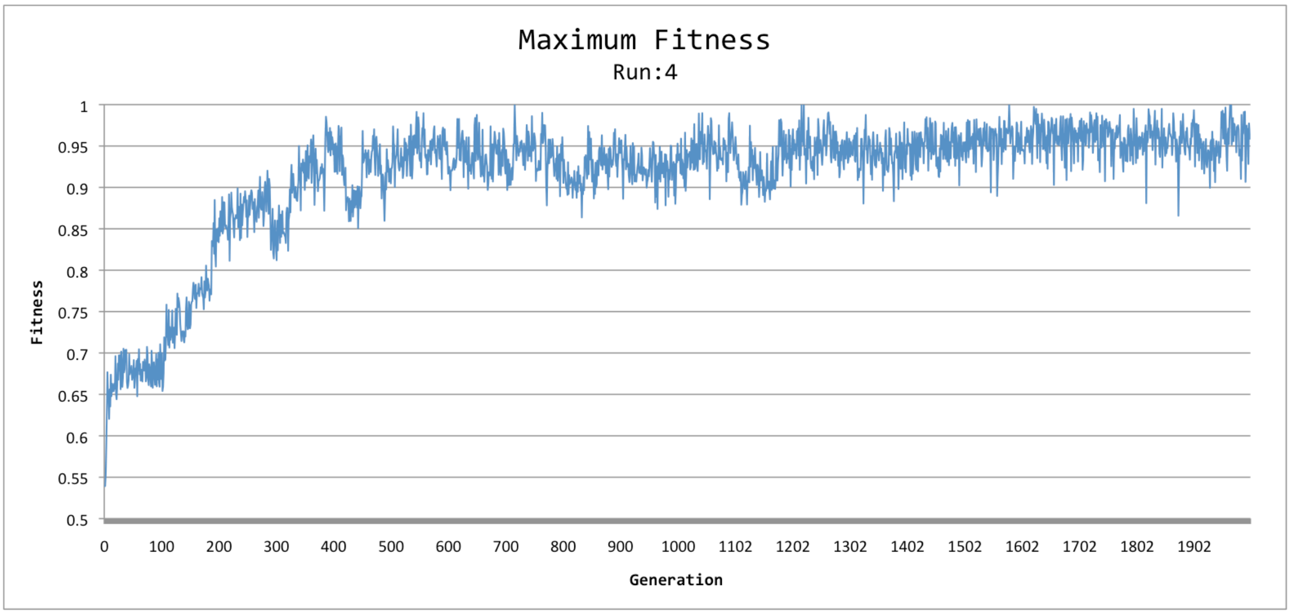

Figure 17. Best fitness in generation (Run 4).

Since classifiers are generated randomly, the behaviour of the agents in the very first generations is purely chaotic. In these early stages the best individuals are the ones capable of showing the widest range of different movements. Pairs that do not move from their initial positions, and pairs that move in an uncoordinated way, get very low fitness scores. Selection pressure pushes towards intermediate behaviours in which the agents tend to stay in sensor range while rotating. The FB behaviour is the first complex behaviour to emerge as a sensory-contact-maintenance strategy around generation 25 (fig.17). Agents producing the FB behaviour are able to move very small distances while still detecting each other thus making them fitter than most of the other agents. The next evolutionary jump takes place around generation 100 where agents start to develop a strategy to maintain the distance to their partners while moving and rotating (Orbit). This new strategy makes the fitness grow very fast from generation 100 to generation 250. Individuals are now able to move their joint centre of mass across bigger distances without being penalized for the loss of contact with their partner. Because the Orbit strategy is by far the most complex behaviour to emerge (3 classifiers involved), it takes lots of evolutionary time to tune it. The amount of different initial positions and angles also makes the Orbit strategy hard to learn. Around generation 300 the different behaviours needed to accomplish the task have already been evolved but need to be integrated. Some individuals are good doing the FB behaviour while others are better at the Orbit strategy. At fitness around 0.9 most of the pairs succeed but there are still some that are not able to accomplish the task. Failures are mostly due to the loss of contact during the last synchronization process (Orbit). At some point one of the agents makes an unexpected move breaking the delicate equilibrium. The first perfect individuals appear in generation 400 with fitness scores greater than 0.975. These individuals are completely in control of the different behaviours (particularly the Orbit) and know when to fire them.

Due to the mutation rate, new individuals with random changes are added to the population at the end of each generation. Sometimes these changes make the agent stronger but sometimes they make it weaker. Since each individual is evaluated against every other individual in the population and fitness is scored in a cooperative way, occasional pairings with the weakest individuals prevent the best individuals from getting a sustained fitness of 1.

4 - Conclusion

In order to argue that ALife models are constructive ways to study the origins of language and communication, we have reproduced the original Quinn (2001) experiment in a different context, and have obtained similar results. Evolutionary runs revealed that coordinated overall behaviour could in fact emerge from a dance-like movement pattern that allowed the two agents to spontaneously establish “leader” and “follower” roles. Since the emergence of a primitive form of communication without explicit communication channels has occurred using a completely different framework, it can be said that Quinn’s results look reasonably general. There is no need for highly sophisticated individuals nor for complex social environments to give rise to primitive forms of signalling. Since one of our main targets is to study the selective pressures towards the emergence of complex forms of language and mind through computational models, the generality of Quinn’s results lets us go one step further.

5- References Cited

Bull, L. (2004), “Learning Classifier Systems: A Brief Introduction”, in Bull, L (Ed.): Applications of Learning Classifier Systems. Berlin u.a, Springer, pp. S. 3 -14.

Calvin, W. H. and Bickerton, D. (2000), “Lingua ex Machina: Reconciling Darwin and Chomsky with the Human Brain”, Cambridge, MA: MIT Press.

Cavanna, A. E., Trimble, M. R., Cinti, F., Monaco, F. (2007), “The ‘bicameral mind’ 30 years on: a critical reappraisal of Julian Jaynes” hypothesis. Functional Neurology, 22, 11-15.

Chalmers, D. J. (1997), “The Conscious Mind“, Oxford University Press.

Dennett, D. (1991), “Consciousness Explained”, Little, Brown & Co..

Dennett, D. (1996), “Kinds of Minds: Towards an Understanding of Consciousness”. Weidenfeld & Nicholson, London.

Dennett, D. (1998), "Julian Jaynes’s Software Archeology." In: Brainchildren: Essays on Designing Minds.

Donald, M. (2001), ”A Mind So Rare he Evolution of Human Consciousness”, Norton, p.202.

Holland, J.H. (1976), “Adaptation”. in R. Rosen & F.M. Snell (eds) Progress in Theoretical Biology, 4. Plenum.

Jaynes, J. (1976), “The Origin of Consciousness in the Breakdown of the Bicameral Mind”, Mariner Books, p.55, p.65, p.66.

Kirby, S. and Lam, L. (2002), “Is emotional intelligence an advantage? An exploration of the impact of emotional and general intelligence on individual performance”, The Journal of Social Psychology, 2002

Noble, J., De Ruiter, J., and Arnold, K. (2010), “From Monkey Alarm Calls to Human Language: How Simulations Can Fill the Gap”. Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems 18, 1, 66-82.

Maynard Smith, J. (1997), “The Theory of Evolution”, C.U.P., 3rd edition.

Nietzsche, F. (1882), “The Gay Science” (Part V, Aphorism No. 354)

Noble, J., Hebbron T., van der Horst, J., Mills R., Powers S. T. and Watson, R. A. (2010), “Selection pressures for a theory-of-mind faculty in artificial agents” ALife XII.

Olin, R. (1999), “Auditory hallucinations and the bicameral mind”, The Lancet, Volume 354, Issue 9173, 10 July 1999, Page 166, ISSN 0140-6736, DOI: 10.1016/S0140-6736(05)75304-6.

Quinn M. (2001), "Evolving Communication without Dedicated Communication Channels", Proceedings of the 6th European Conference on Advances in Artificial Life (September 10 - 14, 2001). J. Kelemen and P. Sosík, Eds. Lecture Notes In Computer Science, vol. 2159. Springer Verlag,London, pp.357-366.

6- Further Reading

Ardery, P. (2004), ”Ramifications of Julian Jaynes's Theory of Consciousness for Traditional General Semantics”, et Cetera.

Bedau, M. A., Packard H., Rasmussen S., Adami C., Green, D. G., Ikegami T., Kaneko, K., Ray T. S. (2000), “Open Problems in Artificial Life”, Artificial Life archive Volume 6, Issue 4, pp.363-376

Bedau, M. A. (2008). “The evolution of complexity”, Thomas Pradeu, et al, “Mapping the Future of Biology: Evolving Concepts and Theories”.

Berlekamp, E., Conway, J. H., and Guy, R. (1982), “Winning Ways for Your Mathematical Plays”, volume 2. Academic Press.

Bureau, P. (2002), “Khepera^2 sensor report”, K-Team

Crutchfield, J. P. (1990), "Reconstructing Language Hierarchies", in Information Dynamics, NATO ASI Series, 256, H. A. Atmanspracher and H. Scheingraber, editors, Plenum Press, New York, pp.45-60.

Crutchfield, J. P. and Young K. (1989), "Inferring Statistical Complexity", Physical Review Letters 63, pp.105-108.

Darwin, C. (1859), “The origin of Species by means of natural selection”, John Murray, London

Dennett, D., McGinn, C., Varela, F., Crick, F., and Penrose, R. (1997), “Explaining Consciousness: The Hard Problem”, Cambridge: MIT Press.

Feldman, D. P., Crutchfield J. P. (1998), “Measures of statistical complexity: Why?”, Phys. Lett. A, 238:244-252.

Fodor, J. (1975), “The Language of Thought”, New York: Thomas Crowell. Gardner, M. (1970) , "Mathematical Games" , Issue of Scientific American, in Martin Gardners

Langton, C. G. (1991), “Life at the Edge of Chaos”, Artificial Life II, SFI Studies in the Sciences of Complexity, vol. X, ed. C. G. Langton, C. Taylor, J. D. Farmer, S. Rasmussen, Addison-Wesley.

McClelland, J. L., Rumelhart, D. E., and the PDP Research Group (1986), “Parallel distributed processing: Explorations in the microstructure of cognition. Volume 2: Psychological and biological models”, MIT Press, Cambridge, MA.

Michel, O. (1996),”Khepera Simulator Version 2.0 User Manual”, http://wwwi3s.unice.fr/˜om/

Noble, J. (2000), “Talk is cheap: Evolved strategies for communication and action in asymmetrical animal contests”, in: From Animals to Animats 6: Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior, pp. 481-490, MIT Press.

Langton, C.G. (1989), "Artificial Life", in Artificial Life, Langton (ed), (Addison-Wesley:Reading, MA) page 1.

Putnam, H. (1961), “Brains and Behavior”, originally read as part of the program of the American Association for the Advancement of Science, Section L (History and Philosophy of Science).

Quinn M. (2001), “A Comparison of Approaches to the Evolution of Homogeneous Multi-Robot Teams”, to appear in Proc. IEEE International Congress on Evolutionary Computation, Korea

Runarsson, T. and Jonsson, M. (1999), “Genetic production systems for intelligent problem solving”, Journal of Intelligent Manufacturing, Vol. 10, pp.181-186.