Abstract—We have studied and developed the behavior of two specific neural processes, used for vehicle driving and path planning, in order to control mobile robots. Each processor is an independent agent defined by a neural network trained for a defined task. Through simulated evolution fully trained agents are encouraged to socialize by opening low bandwidth, asynchronous channels between them. Under evolutive pressure agents spontaneously develop communication skills (protolanguage) that take advantages of interchanged information, even under noisy conditions. The emerged cooperative behavior raises the level of competence of vision guided mobile robots and allows a convenient autonomous exploration of the environment. The system has been tested in a simulated location and shows a robust performance.

Keywords: Evolutive robotics, robotic vision, neural reactor, cooperative agents, neural nets, agent’s communication.

I. INTRODUCTION

To reach the degree of complexity required by operative agents in mind theory [1] or agent theory [2], neural networks seem to be good nominees. For this to happen, however, one of the problems to be solved is how to communicate and induce cooperative behaviors in fully trained neural processors.

Perhaps one of the reasons why artificial neural nets have not yet reached the outstanding processing power of their biological cousins is their lack of integrating tools. Current artificial neural networks provide devices that cover a wide range of practical applications, such as pattern recognition, image compression, route planning, and others. It is not clearly established, however, how these fully trained neural tools, once debugged and tested, can be made to cooperate in order to obtain more advanced neural processors.

In related papers simulated evolution has been used in neural training [3] and neural structuring [4]. We have used it to make commanding, independent, fully operative neural agents develop communication skills and cooperate in the task of controlling populations of mobile robots. The search develops around two specific kinds of neural processors:

- Vehicle Driving Agents or VDAs use vision devices as parallel input. They are constructed with standard DC sigmoid neurons and trained with back-propagation.
- Route planning agents or RPAs use problem related synaptic information as knowledge and are constructed with selectively-inhibited Hopfield neurons that undergo cycles of high and low energy.

Vehicle Driving Agents based upon neural nets have been studied in automatic vehicle driving [5] and robotic vision [6]. Route planning agents defined by selectively-inhibited Hopfield neural nets, subjected to collective cycles of low and high energy, have been used to solve route planning problems, such as the Travelling Salesman Problem (TSP) and others [7].

The basic structure in this design philosophy is the “n-flop” proposed by [7] in his methodology of “programming with neurons”.

The n-flop structures add the advantages of having a robust behavior under partial network failures [8] and the capacity of solving high dimensional problems [9]. We use n-flops as self sufficient “neural reactors” which consume “energy” (or processing time) and produce a random but unique state in each firing process.

This study shows how independent vehicle driving agents (VDA) and route planning agents (RPA) can evolutively learn to communicate through low bandwidth channels and raise the level of competence of mobile robots guided by composite eyes. This happens in a layered oriented control architecture where the route planning agent (RPA) assumes the “level zero control” of the robot while the other agents cooperate as modulating controllers [10].

Complexity, an omnipresent ingredient in natural cognitive systems [11], is an integral part of contemporary neural design philosophy [4]. The proposed method incorporates the benefits of complexity by promoting the merging of fully trained neural agents, which makes it possible for many possibilities and variations to show up.

Minds also require a high richness of dynamic states [11]. We encourage this mechanism by allowing complex noisy information coming from a vehicle driving agent (VDA) to influence the cooling phase of a route planning agent (RPA) that is trying to reach an equilibrium state.

II. ROBUST LAYERED CONTROL

The control problem for the robot is partitioned considering levels of competence; quoting from [10]:

“A level of competence is an informal specification of a desired class of behaviors for a robot over all environments it will encounter. A higher level of competence implies a more specific desired class of behaviors.”
In [10] each level of competence is handled by a debugable finite state machine that produces a class of behavior in the robot. In other words, once the robot has a reliable controller for a given level of competence, the robot will do something in whatever place it is located or even if abandoned. In order to move to a higher level of competence, another finite controller must be designed, debugged and added to the existing one through a level of competence coupling protocol called subsumption [10]. In this arrangement, a key condition for robustness is that the communication between controllers has to be made through low bandwidth channels, since otherwise control will be plagued by contradictions [10], [12].

For this paper, level of competence ZERO is achieved by using a route planning agent (RPA) $A_0$, that undergoes cycles of high and low energy and handles the robot moving hardware directly. This agent behaves as a self sufficient “neural reactor” which consumes energy and produces a random sequences of equilibrium states. One of the experimental results of this paper shows that these forthcoming sequences can be statistically modulated by external events. Once agent $A_0$ proves to be reliable and is thoroughly debugged, it remains unchanged and serves as the embryo for a cooperative society of agents, where higher levels of competence are pursued. In its primal phase, $A_0$ does not communicate with any other robot control agents, yet its neural reactor, acting as a compulsive behavior initiator, induces in the robot a class of behavior known as random walk [13]. The pursuing of random movement in isolated conditions confers the proposed route planning agents (RPA) their “proactiveness” [2]. Cooperation is promoted by defining sets of connecting biasing weights between proactive agent $A_0$ and other participant vehicle driving agents (VDA), which behaves as image compressor and concentrate available image information into the output of a few neurons.

### III. Method

The output of a well trained vehicle driving agent is connected as input to an autonomous route planning agent $A_0$, through an organized map of biasing weights. Simulated evolution is used to polish this communication channel until the level of competence of the controlled mobile robot raises from pure random walk to a walk with vision assisted task-achieving features.

#### A. The Robot

Figure 1 describes the virtual robot used in this paper (top view), which is a tricycle type one, equipped with two drive wheels at the back and one swivel caster wheel at the front. It also has two fixed composite eyes pointing forward, with 180 degrees field of view. The robot contains enough hardware/software so that displacement is controlled by assigning speeds to the drive wheels. The robot travels in a 2-D area where other elements are present. For the time being only one robot is considered.

#### B. The environment

In figure 1 short round objects (dots) represent metal “nuggets” and the robot should move forward and capture them. Longer objects (rectangles) represent “loose structures” where the robot may get entangled and perish.

The robot is required to accomplish the process NUGGET GATHERING defined as: Move freely and autonomously in the field, capture as many nuggets as possible, and stay away from rectangles. Under certain angles of vision rectangles may look as nuggets or bunches of nuggets may look like rectangles. In any case the agents responsible for robot control, sooner or latter, will have to deal with uncertainties about the meaning of some specific images.

At the operative level, robots that touch rectangles perish and are taken out of the evolutive trail.

#### C. The Vehicle Driving Agent

For this paper a vehicle driving agent $A_1$ is defined by a trained neural net, with two composite eyes that feed two layers of sigmoidal neurons, with 15 hidden and 5 output elements respectively (fig. 2).

For computer simulation, composed eyes have 30 binary ommatidia each and are separated by an equivalent of two eye diameters to allow the triangulation of the objects. Ommatidia have a maximum reach of five robot body length. Every composed eye can visualize $2^{30}$ or around $10^{10}$ different images.

Before it reaches the category of agent a VDA has to go through a finite process called "training", which in our case is done through back-propagation [5] and by using the training set in fig. 3, prepared by skilled humans.

A successfully trained VDA develops inference capacity and acts as an “image compressor”, where high dimensional image information is compressed into the output of a few neurons [5], [6]. In our case, the chosen VDA is selected to compress all image activity ($10^{10}$ possible images) into the output of five neurons associated with one out of five arbitrarily chosen basic robot movements. By definition this VDA does not produce real movements in the robot but only provides compressed image information to modulate the random behavior of proactive agent $A_0$. 

![Fig. 1. Two dimensional virtual mobile robot with composite eyes](image-url)
Fig. 2. Vehicle Driving Agent (VDA). It is formed with 15 hidden neurons and 5 output neurons. Once trained, all images perceived through composite eyes are compressed in the output of five neurons.

Once properly trained $A_1$ produces compressed, valid but noisy information concerning robot expected behavior.

The Network is trained as a classifier so that the potential robot stays away from rectangles and comes close to the nuggets.

Each training example contains one image for each composite eye and a vector target $T = (T_1, T_2, T_3, T_4, T_5)$ that indicates the target value for each neuron output. Being trained as a classifier [5], only one target at a time will be on (1) while all other four will be off (0).

For instance, if the image shows a facing rectangle in front (the second row of training set in figure 3), the targets should be 00100, meaning that the third neuron should be “on” so that the robot moves backward (in real training 1’s are substituted by 0.9 and 0’s by 0.1).

When successfully trained, the proposed VDA behaves well in clearly marked conditions, such as isolated images of nuggets or rectangles. We tested these neural agents under many different parameter variations (neurons gain, biasing, number of hidden neurons, number of out neurons etc.), and all the solutions that were found show more or less the same innate handicap: lack of proactiveness when complex images, corresponding to a mixture of nuggets and rectangles, have to be processed.

D. The Level Zero Route Planning Agent

The responsibility to move the robot at zero level of competence is assigned to a five state random machine called agent $A_0$. This agent (fig. 4) defined by a 5-flop, is carefully tuned to behave as a random state machine. To this end, every neuron $a_i$ inhibits all other four neurons with weights -1. The common excitatory input K acts as a firing source by following a cyclic ramp activity with values 0.01 to 6.5. Internal neural gain for every neuron is fixed in 20. With each ramp in K the 5-flop fires and produces a unique random winner (equilibrium state). Through non intelligent buffers, this winner works as a behavior initiator that triggers one out of five basic behaviors in the robot. Each basic behavior corresponds to assigning fixed speed to each robot wheel, as denoted in the fig. 4.

With no connections with the outside world and responding to a rapid sequence of firing pulses , $A_0$ induces in the robot a kind of wobbly, energy consuming random walk (figure 5). This represents zero level of competence for the robot. Notice that the resultant zero level control agent $A_0$ is intensively proactive and will try to keep the robot moving even if it is isolated from any other source of information in the robot universe. In control terms, $A_0$ gives the robot the power to move around without guidance or sensory information. It is later shown that this primal, energy consuming navigation conduct evolves upward when $A_0$ is encourage to use sensory noisy guidance information coming from other agents.

IV. AGENTS’ COOPERATION

Figure 6 shows an arrangement where intensively proactive agent $A_0$ makes contact with vehicle driving agent $A_1$ who acts as a trained sensory device that provides noisy driving
Fig. 5. When controlled by the isolated route planning agent $A_0$, the mobile robot executes a wobbly random walk. This corresponds to the level zero of competence of the robot. Robot can maintain this behavior even if it is completely private from sensoring information.

Fig. 6. Cooperative Control Agent formed by a vehicle driving agent $A_1$ (upper left) and level zero route planning agent $A_0$ (lower right). The 25 biasing weights are codified in a 25 genes chromosome, 8 bits per weight, giving origin to a search space with $2^{200}$ elements.

information. The objective is to allow the intensively proactive agent $A_0$ to extract useful sensory information from noisy agent $A_1$, so that the overall robot control system raises its level of competence. Communication takes place through a low bandwidth asynchronous input channel defined by 25 biasing weights. Driving information provided by $A_1$ is used asynchronously by $A_0$ only during its cooling phase.

If all biasing weights are set to zero, the robot returns to level zero of competence, that is, to walk randomly.

If biasing weights are modified, the kinetic behavior of the robot changes. In theory, there may exist an appropriate set of biasing weights that creates a cognitive association between vision and route planning, and raises the robot’s level of competence. This step is solved by using a genetic algorithm.

A. The Genetic Algorithm

Genetic algorithms are search algorithms used to find near-optimal solutions in arbitrarily created search spaces. In this work, the search space is given by Biasing Maps, which encompass the set of all possible values assumed by the 25 biasing weights between agents $A_1$ and $A_0$.

The employed genetic algorithm [14] consists of five key operations:

1) Chromosome definition: In figure 6, the 25 biasing weights are set to range from +2 to -2 and are codified in eight bits genes each. This produces a chromosome (genotype) with 25 genes and 200 bits. Using this chromosome, the genetic search space is restricted to $2^{200}$ points.

2) Selection: is carried out using the “slide window” method, where the probability of a genotype to be selected for mating is made proportional to its fitness value by using a fitness-sorted list of participants [15].

3) Crossover: two parent chromosomes selected for mating interchange genetic information in a gene-by-gene basis. For each gene a random cutting place is selected and the resulting gene segments are interchanged. The resulting elements turn into child chromosomes.

4) Mutation: is implemented by iterating all bits in the chromosome and switching the value of each randomly. The probability of changing one bit is called the mutation rate and is here maintained in 10%.

5) Fitness: is assigned by evaluating in the field the displacement behavior of each robot using a task-achievement reward method. Each robot is located in generic situations and its internal agent $A_0$ is fired $p$ times, with $p$ being a random number experimentally fixed between 7 and 25. Three desired attributes are sought:

- Forward displacement skill. Robots that move forward frequently receive good grading. This saves energy, reduces wobbly behavior and increases mean covered distance. This attribute is measured by counting, in $p$ firing, how many times – denoted by $p_f$ – the forward neuron wins. That is,

$$F_1 = \frac{p_f}{p}$$  \hspace{1cm} (1)

- Nugget gathering skill. Robots that properly move toward nuggets receive good grading. This is measured by counting, in $p$ firing, how many times – denoted by $p_n$ – the robot that has nugget in its field of view comes closer to it. i.e.

$$F_2 = \frac{p_n}{p}$$  \hspace{1cm} (2)

- Rectangle evasion skill. Robots that properly move away from rectangles receive good grading. This is measured by counting, in $p$ firing, how many times – denoted by $p_r$ – the robot that has a rectangle in its field of view moves away from it. i.e.

$$F_3 = \frac{p_r}{p}$$  \hspace{1cm} (3)

The total fitness function is a linear combination of this evaluation:

$$F_T = \frac{F_1 + F_2 + F_3}{3}$$  \hspace{1cm} (4)

This fitness is assigned value zero if, in $p$ firing, the robot gets in touch with a rectangle. Values of $p$ vary randomly from 7 to 25. The genetic algorithm that is employed [14], [15] uses a population of 50 chromosomes and can be written in pseudo code as follows:
Fig. 7. Evolved nugget gathering journey. Under evolutive pressure, agents $A_0-A_1$ learn to cooperate and random walking develops into an autonomous, organic task-oriented walk, which utilizes visual information. This makes possible a quasi-optimal visual search inside a dangerous and intricate scenario.

Choose initial random population
Evaluate each individual’s fitness
Repeat
  Select, proportional to fitness, two individuals to reproduce
  Apply crossover operator
  Apply mutation operator
  Evaluate each individual’s fitness
  If a new population is completed:
  discard old population and advance one generation
Until terminating condition

V. RESULTS

The algorithms have been developed in Borland C++ and run in different Windows versions.

Figure 7 shows an evolved high ranked composite cooperative agent working in a typical nugget gathering journey. The composite agent utilizes the intensively proactive agent $A_0$ in order to move the robot in an organic route devoted to capture available nuggets. The described route shows its random heritage but thanks to the information supplied by $A_1$ the robot successfully avoids rectangles while keeping a quasi optimal track on nuggets.

This evolved emergent behavior indicates that agent $A_0$ makes good use of the noisy sensory visual information coming from agent $A_1$.

Interesting enough high ranking agents will keep moving on the robot, in a spontaneous random search, even if all images are retired from their field of view (blind search) or if a random image is fixed in the robot eyes (vision failure).

Figure 8 shows evolved complex phenotypes formed by strings of twenty five biasing weights. Results are shown for top ranked individuals at different evolutive trials. Weights range from -2 to +2 and for graphic purposes are codified in gray/black concentric circles. Figure 9 shows typical fitness improvement for top individuals during evolutive time.

VI. CONCLUSIONS

As an original contribution we show a design method where artificially evolved low bandwidth channels connect fully trained neural agents and produce robust, self willed, vision-guided robots.

The fact that high ranking agents keep on moving the robot when no sensory input is available indicates a strong emergent proactivity, where sensory information modulates rather that determines the system behavior.

The evolved system tolerates noisy information and gives the robot the power to “move around by using noisy guidance” It is so because under certain angles of vision rectangles may look as nuggets or vice-versa. Uncertainties thrive in circulating image information and yet the robot travels successfully through crowded scenarios.

Most times robots do not execute a perfect search but just a near to optimal one. This behavior is shared with most brained animals.

That the robot increases in competence is an indication that circulating information makes sense and that an agent “protolanguage” has emerged.

As defined, evolved control structure is a form of modulated subsumption where agent $A_1$ subsumes agent $A_0$. In this sense there may evolve another agent $A_2$ that subsumes $A_0$, $A_1$, and creates a higher level of competence, etc.

If biasing weights between $A_1$ and $A_0$ are collectively activated or deactivated, then the whole process of NUGGET GATHERING will be remembered or forgotten. This effect is called k-line and is considered an important type of agent in mind theory.

Since they all share the known artificial neural net technology, proposed neural agents present a good profile in
ACKNOWLEDGMENT

The author would like to thank: The Universidad Politécnica de Madrid (UPM-ETSII) for academic and financial support. Prof. Rafael Aracil for advising and technical guidance. Prof. Pascual Campoy for proofreading, valuable suggestions and sharp ideas. Prof Alexandre Campos for supporting technical discussions and appreciated editorial help. The graduate students in the Robotic Vision Group (UPM-ETSII) and the robotic grad course in the Universidad Central de Venezuela (UCV) for enduringly listening to new ideas.

REFERENCES