Algoritmo evolutivo para hardware evolucionable

An evolutionary algorithm for evolvable hardware

Celso De La Cruz[1], PhD, Teodiano F. Bastos[1], PhD, Héctor D. Patiño[2], PhD, Ricardo Carelli[2], PhD
1. Depto. de Ing. Eléctrica, Universidade Federal do Espírito Santo, Vitória, ES, Brazil
2. Instituto de Automática, Universidad Nacional de San Juan, San Juan, Argentina
celsodelacruz@gmail.com; tfbastos@ele.ufes.br; dpatino@inaut.unsj.edu.ar; rcarelli@inaut.unsj.edu.ar

Recibido para revisión 16 de marzo de 2009, aceptado 20 de mayo de 2009, versión final 28 de mayo de 2009

Resumen**—**Este trabajo presenta un análisis de la evolución de especies en la naturaleza. A partir de este análisis, algunas propiedades evolutivas son extraídas las cuales son consideradas para el diseño de un algoritmo evolutivo para hardware evolucionable. Estas propiedades reducen el riesgo de mal funcionamientos en el sistema físico cuando está evolucionando. El algoritmo propuesto es diseñado especialmente para evolucionar hardware, lo que significa, mediante el uso de métodos computacionales evolutivos, los sistemas mecánicos y eléctricos son automáticamente diseñados, adaptados, y reconfigurados. Fueron realizadas simulaciones y una experimentación para evaluar el desempeño del algoritmo evolutivo.

Palabras Clave**—**Algoritmos Evolutivos, Hardware Evolucionable, Sistemas Inteligentes, Métodos de Optimización, Sistemas Robóticos, Evolución de Especies.

Abstract**—**This work presents an analysis of the evolution in nature of species. From this analysis, some evolution properties are extracted which are considered to design an evolutionary algorithm for evolvable hardware. These properties reduce the risk of malfunctions in a physical system when it is evolving. The proposed algorithm is designed specially to evolve hardware, i.e., through the use of evolutionary computation methods, the electrical and mechanical hardware systems are automatically designed, adapted, and reconfigured. Simulations and experimentation were carried out in order to evaluate the performance of the evolutionary algorithm.

Keywords**—**Evolutionary Algorithms, Evolvable Hardware, Intelligent Systems, Optimization Methods, Robot Systems, Species Evolution.

I. INTRODUCTION

The evolution process experienced by a species in nature turns their members more efficient as regard survival in a changing and complex environment. These processes can be compared to the minimization (or maximization) of a cost function in a mathematical optimization problem. Numerous works on evolutionary algorithms have been presented that are based on the theory of natural evolution [6], such as Evolution Strategies (ES), Evolutionary Programming (EP) and Genetic Algorithm (GA). Although these Evolutionary Algorithms are widely applied to many fields, the mathematical foundation of evolutionary algorithms is very weak. There has been growing interest in a mathematical model to character the evolutionary algorithm. In [11], an abstract mathematical model is presented, where it uses an abstract selection and evolution operators.

Evolvable hardware [10] is an exciting and emerging field. It lies at the intersection of evolutionary computation and physical design. Through the use of evolutionary computation methods, the electrical and mechanical hardware systems are automatically designed, adapted, and reconfigured. There are many researches in this field. For example, in [12] a field programmable gate array was evolved to discriminate a tone, in [9] an evolutionary system to automatically design both the robot morphology and controller was developed, in [7] an autonomous evolution of a dynamic gait on Sony's entertainment robot was developed, and in [2] a normalized steady state genetic algorithm is proposed and used to design by simulation a ST5 antenna (X-band antenna for NASA´s Space Technology 5 spacecraft).

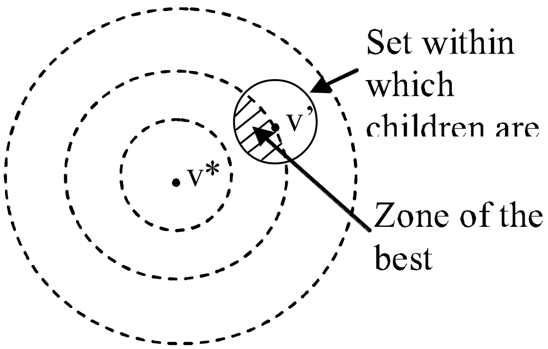

Figura 1. Way to evolve a species. The points v' and v* are, respectively, the vector of characteristics of the fittest individual and the optimal individual

Space Technology 5 spacecraft).

The present work considers the fact that almost all animals of a species in nature are able to survive and only a few animals die because of malfunctions in their organisms. The aim of this work is to obtain similar properties when evolving hardware. On the basis of the properties of the species in nature, a new computational evolutionary algorithm is developed. The proposed algorithm is characterized by gradual changes in the mathematical representation of the species at the time of evolution. In addition, this algorithm is also characterized by showing a bounded mathematical representation of the species within a parameters space. The last two properties do not help the evolution to escape from a local minimum. However, they give good convergence properties and are good for evolvable hardware, because they can make the risk of malfunctions of the real system null during the evolution. The evolutionary algorithm was tested on an Ackley function, in order to show some characteristics of the evolutionary algorithm and to compare with other evolutionary algorithms. Finally, an experiment was performed using a unicycle-like mobile robot which has a tracking control system. This experiment consisted of letting some control parameters (dynamic model parameters of the robot) of the control system evolve by themselves while it executes a repetitive task.

II. EVOLUTION OF A SPECIES

A. Evolution in nature

The animals of a species in nature share similar characteristics. If these characteristics could be put in a vector space, a set of the animal characteristics vectors of the species would be grouped in a zone. In many cases, the evolution of a species in nature is gradual [3] and it would take many generations for a noticeable change to take place.

From the sexual selection studied in [3], the following can be concluded: the number of children in a group of parents will show greater probability of having a large value when those parents are more similar in characteristics to the fittest parent. Conversely, the number of children in a group of parents will have greater probability of having a low value when those parents differ too much in characteristics from the fittest parent, and/or differ too much in characteristics from each other. The last analysis considers the difficulty in bearing children between two biologically different animals (different with respect to their characteristics). Indeed, they could be incompatible to have descendants and, consequently, will have no descendant with probability 1. The mutants, deformed or mutilated children have extremely different features from the animals of their own species, which makes them most unlikely to have children. Even if these animals were capable of reproducing, they would most likely be rejected for reproduction by the members of their own species.

B. Evolution of a physical system

Let us consider three things: 1) an individual is a physical system (a robot, an industrial plant, etc.), 2) the species' individuals have similar characteristics, and 3) the evolution of this species is gradual. Then, the following analysis is done. If the physical system (individual) of a species works well, then the physical systems of the same species sharing similar characteristics will work well too with high probability. Hence, the risk that the physical system breaks down in an evolutionary process is very low or null. The eliminated individuals are those that do not work as well as the rest of the group; those individuals eliminated do not necessarily work badly. There is a great probability that some children will work better than their parents, consequently, the species will have a tendency to a minimum of the cost function.

C. Example of evolution

For example, let us consider an evolution process with a convex function $F(x)=\|x\|^2$, and a uniform random distribution in the mutation operator. In this example, if the children dispersion is small enough, the probability that the fittest individual creates a child with less cost than that of itself is in a range close to 0.5 (see Fig. 1).

III. DEFINITIONS

The following definitions are used in the mathematical analysis of the evolutionary algorithm proposed in the next section.

A. Theoretical definitions

1) Power or Parts of a Set: The set of all subsets belonging to $B \subseteq X$, where X is a universal set, is called the power of B and is denoted as $P(B)$ or 2^B. $P(B)=\{A \subseteq X: A \subseteq B\}$.

2) Metric Space: A metric space is a pair (X,d), where X is a set and d is a metric in X (or a distance function on X), that is, a function defined on $X \times X$ such that for all $x,y,z \in X$ we have:

M1: d is finite, non negative and real-valuated

M2: $d(x,y)=0$ iff $x=y$

M3: $d(x,y)=d(y,x)$

M4: $d(x,y) \leq d(x,z)+d(z,y)$.

3) Vector Space: A vector space (or linear space) over a field K is a nonempty set X of elements $x,y,\ldots$ (called vectors) together with two algebraic operations. These operations are called vector addition (that is a mapping $X \times XX$ which is commutative and associative) and multiplication of vector by scalars (that is a mapping $K \times XX$ which is distributive). To delve into this subject refer to [8].

4) Banach Space: A Banach Space is a complete normed space (complete in the metric defined by the norm). A normed space is a vector space with a norm defined on it (to delve into this subject refer to [8]).

5) Ball: Given a point $x \in X$, where X is a metric space, and a real number $r>0$, two types of sets are defined:

- $\breve{B}(x;r)=\{y \in X : d(y,x)<r\}$ (Open ball)
- $\tilde{B}(x;r)=\{y \in X : d(y,x) \leq r\}$ (Closed ball)

6) Neighborhood: A neighborhood of $x \in X$ is any subset of X, where X is a metric space, that contains an open ball $(x;\varepsilon)$.

7) Global and local minimum on a wide sense: Let the function $f:X \to R$, where X is a metric space and R is the real line. The global minimum, on a wide sense, of function f is the point $x^* \in X$ such that $f(x^*) \leq f(x)$ for all $x \in X$ different from point x^*. The local minimum, on an wide sense, of function f is the point $x^\wedge \in X$ such that $f(x^\wedge) \leq f(x)$ for all $x \in E^\wedge$ different from $x^\wedge$, where $E^\wedge$ is a neighborhood of $x^\wedge$.

B. Definitions formulated to be used in this work

1) Individual: It is a biological animal, or a physical system, or software, or any other system having a defined vector of characteristics. This vector has information on the features that differentiate the biological animals, or physical systems, or software, etc. The characteristics in a biological animal may be: size of adult animal parts, adult muscular mass, adult brain volume, etc. The characteristics in other systems may be: control gains, parameter values, etc. When a vector of characteristics of a physical system is changed, a new individual is obtained. When we are working in a vector space, the individuals are represented by its vector of characteristics.

2) Creation of a Child: It is the process of mutation and/or recombination of the vector of characteristics of one or several individuals to attain a new vector of characteristics. The child, then, is the physical system with this new vector of characteristics.

3) Fitness: The extent to which an individual is adapted to or able to produce children.

4) Species: It is a group of individuals that share a common ancestor (a parent or, recursively, a parent of an ancestor), a lineage (a common ancestor) that keeps its integrity with respect to other lineages, both in time and space. At some point in the progress of the group, the members may diverge into two branches. When such a divergence becomes clear enough, each new branch is considered a different species.

5) Species dispersion: A term that refers to the location, in a vector space, of species' individuals related to a central value.

6) Space S: It is a Banach space, whose elements or points are characteristic vectors of individuals. These individuals possess an equal structure (same number of limbs, similar organs, with feathers or hair, same physical structure, same software structure, and the like).

7) Cost functional F and cost of an individual: cost functional F is a functional with domain in the space S and range in the positive real numbers; thus, $F:S \to R^{+}$. This function assigns a cost to each individual. The costs functional F must be such that $F(x) \to \infty$ with $\|x\| \to \infty$, where $\|x\|$ is a norm of x [8].

8) Fittest Individual: It is the individual which has less cost than the other individuals of the species, or which is the most adapted to or the ablest to produce children.

9) Optimal Individual: It is the individual which has a vector of characteristics equal to the global minimum, on a wide sense, of function F.

10) Evolution stage or generation: It is the average time interval between the creation of parents and the creation of their children.

11) Death by inefficiency: It is the elimination, based on some elimination criteria, of an individual which has high costs with respect to other individuals of his same group.

IV. EVOLUTIONARY ALGORITHM (M,1+Λ)-E1E

In the evolutionary algorithm $(\mu,1+\lambda)$-E1E, the term $(\mu,1+\lambda)$ means that the μ parents will have λ children. Then, the λ children and the fittest of the μ parents will be evaluated in order to obtain μ individuals with the smallest cost values to form the new population of the species. E1E indicates the evolution of only one Species.

In the algorithm $(\mu,1+\lambda)$-E1E, it should be complied with $(\lambda+1) \geq \mu$, $\mu \geq 2$ and $\lambda \geq 2$. The first condition is aimed at ensuring that the new population has μ individuals. The second and third ones are necessary for self-adjustment of the species' dispersion, which will be covered further below in this section.

A. Algorithm (m,1+l)-E1E

The procedure is as follows:

1. Initialize the first individual a_1^0, then initialize a^0 such that $a_i^0 \in \tilde{B}\left(a_1^0; r/c_0\right)$; $i=2,...,\mu$..

2. Determine the fittest individual v'^{0} and initialize $n=1$.

3. While the finalization criteria are not yet met, perform the following:

a. Select $\bar{\mu}^{n-1}$ parents. These selected parents are the elements of a^{n-1} contained by the closed ball $\tilde{B}(v'^{n-1},r)$.

b. Modify the order of a^{n-1} such that:

$$a_i^{n-1} \in \tilde{B}\left(v'^{n-1}, r\right), \text{ for } i = 1,...,\bar{\mu}^{n-1}; \text{ and}$$

$$F\left(a_1^{n-1}\right) \leq F\left(a_2^{n-1}\right) \leq ... \leq F\left(a_{\bar{\mu}^{n-1}}^{n-1}\right)$$

c. If=1, create the children in the following way:

$$Z_{1k}^n = v'^{n-1} + X_{1k}^n r / c_0$$

where $1 \leq k \leq \lambda$.

If $\bar{\mu}^{n-1} \geq 2$, create the children in the following way:

$$Z_{jk}^n = v'^{n-1} + X_{jk}^n d\left(a_j^{n-1}, v'^{n-1}\right) / c_0$$

where $j \geq 2$, and Z_{jk}^n is the k-th child created using the first and j-th selected parent. The number of children created by the first and j-th selected parent is calculated as follows:

$$\bar{\lambda}_j^n = \begin{cases} \lambda - fix\left(\dfrac{\lambda}{\bar{\mu}^{n-1} - 1}\right)\left(\bar{\mu}^{n-1} - 2\right) & \text{if } j = 2 \\ fix\left(\dfrac{\lambda}{\bar{\mu}^{n-1} - 1}\right) & \text{if } j = 3, 4, .., \bar{\mu}^{n-1} \end{cases}$$

d. Form the vector b^n with all the created children in the following way:

$$b^n = \left(b_1^n, ..., b_l^n, ..., b_\lambda^n\right) \in S^\lambda; b_l^n = Z_{jk}^n \in S$$

e. Calculate a^n:

$$a^n = T_2^n\left(a^{n-1}, b^n\right).$$

f. Increment $n=n+1$.

In the algorithm, the closed ball $\tilde{B}(.,.)$ is $\tilde{B}(x,r) = \{y \in S : d(y,x) \leq r\}$; $d(.,.)$ is the metric induced by the norm of space S; r and c_0 are real constants (algorithm parameters); $fix(.)$ is a function that returns the integer part of its argument; $X_{jk}^n \in S$ is a random variable having uniform probability on the closed ball $\tilde{B}(0,1)$; the individuals a_i^n, $i = 1, 2, ..., \mu$ form the vector $a^n = \left(a_1^n, a_2^n, ..., a_\mu^n\right) \in S^\mu$;

and v'^n is an element of a^n such that

$$F\left(v'^n\right) = \min_i\left(F\left(a_i^n\right)\right).$$

Mapping $T_2^n : S^\mu \rightarrow S^{\mu+\lambda}$ evaluates v'^{n-1} and the elements from b^n, selects m characteristics vectors which give the smallest cost values and forms a vector with the m selected vectors.

smallest cost values and forms a vector with the μ selected vectors.

Possible spaces S can be: a real vector space with 1-norm; a real vector space with 2-norm, or a real vector space with ∞-norm (these three are Banach spaces).

B. Self-adaptation of the species dispersion

In the creation of children, the distance $d\left(a_j^{n-1}, v'^{n-1}\right)$ is used, which is the distance between v'^{n-1} and another element of a^{n-1}. Constant c_0 can be taken as $c_0 = \sqrt[N]{P_{co}}$ if considering S equal to the real vector space of dimension N with ∞-norm. This constant makes $\left(X_{jk}^n / c_0\right) \in \tilde{B}(0;1)$ be complied with a probability P_{co}. Therefore, it is expected that $100 \times P_{co}$% of the children will be less dispersed than the parents, and the remaining children will be more dispersed than the parents. The dispersions are measured with respect to . The self-adaptation of the species dispersion can be controlled by the parameter P_{co}.

Choosing Pco=0.5, the dispersion of the species tends to increase when such species is relatively far from a minimum of the costs function (far with respect to their distance on space S). The reason is that the most distanced children to the fittest one of their parents will have greater probabilities of having the smallest cost of the species than the least distanced children. Figure 2 shows this fact when using the same example of section 2.

Conversely, choosing again Pco=0.5, the dispersion of the species tends to decrease when such species is relatively close to a minimum of the costs function, because, now, the least distanced children to the fittest one of their parents will have greater probabilities of having a smaller cost than the most distanced children. Figure 3 shows this fact when using the same example of section 2 again.

Only some special cases do not have these properties. One example is illustrated in Fig. 4. That would happen in this example is a decrease of the species dispersion while it go through the strait zone.

This self-adaptation of the dispersion of a species increases

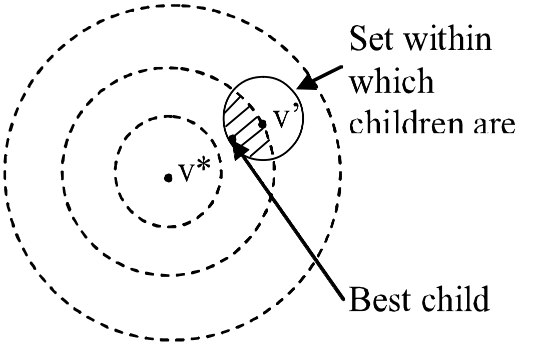

Figure 2. Best child when the species is relatively far from the minimum point. The points v' and v* are, respectively, the vector of characteristics of the fittest individual and the optimal individual.

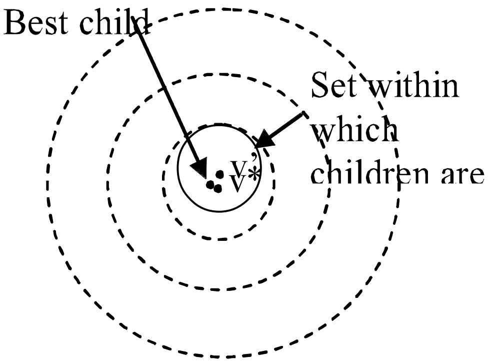

Figure 3. Best child when the species is relatively close to the minimum point. The points v' and v* are, respectively, the vector of characteristics of the fittest individual and the optimal individual.

C. Making null the risk to break down a physical system

Let us consider two zones in space S: The first one, the *Bad Zone,* whose points are characteristics vectors of individuals prone to collapse (for example points that have costs equal or greater than 60 in Fig. 5). This means that the physical system works very badly or that it has a high risk of breaking down. The second, the *Acceptable Zone*, whose points are characteristics vectors of individuals that acceptably perform their task (for example points that have costs equal or less than 50 in Fig. 5).

The dispersion of parents that will have children is limited by the closed ball $\tilde{B}\ (v'^{,n-1},r)$. Considering this fact, the vector of characteristics of the children will be placed within the closed ball $(v'^{,n-1};r/c_n)$, on account of the nature of the probability density function with which the children are created.

The aim here is to avoid the characteristics vector of a child getting into the Bad Zone. Algorithm (M,1+Λ)-E1E accepts only a vector v'^{n} with equal or less cost than that of v'^{n-1}. Let us consider that algorithm (M,1+Λ)-E1E begins with within the Acceptable Zone. Every vector v'^{n} with $n \geq 0$ is equal to or equal to a vector with a smaller cost; therefore, v'^{n} is in the acceptable zone $\forall n \geq 0$. Then, if r/c_n has low enough value, the vector of characteristics of a child will never get into the bad zone. For example, let us consider $r/c_n=0.1$, then, the distance between the characteristics vector of a child and v'^{n-1} is smaller than or equal to 0.1. If we consider Fig. 5 in this example, the characteristics vector of any child will never get into the bad zone; because, to happen, the distance between the vector of characteristics of the child and v'^{n-1} has to be equal or greater than 0.2. Therefore, the constant r has to be defined with a value that makes null the risk to break down a physical system. The

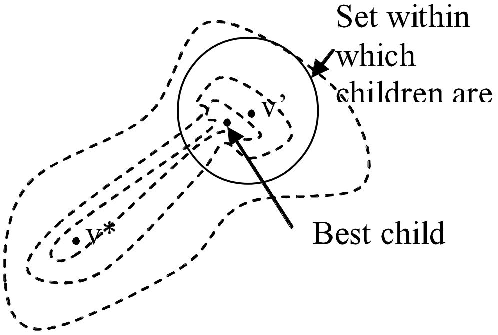

Figure 4. Special case of self-adaptation. The points v' and v* are, respectively, the vector of characteristics of the fittest individual and the optimal individual

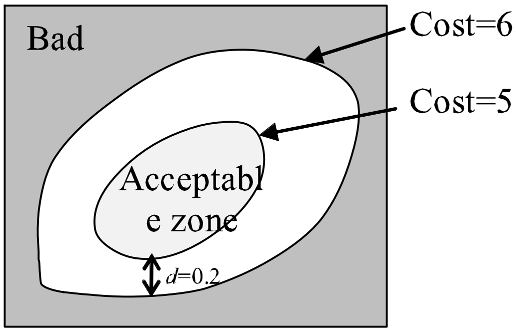

Figure 5. Graph of Acceptable and Bad values of the characteristics vectors of individuals. Here d is the minimum distance between points of acceptable and bad zones.

value of r can be defined based on simulation results. If the mathematical model does not accurately describe the real system, a smaller value of r than that defined by simulations can be used.

In Sections 4.2 and 4.3, the properties of algorithm (μ,1+λ)-E1E are reviewed. These properties improve the performance of the evolutionary algorithm (i.e., good convergence rate and better chances of implementing it on evolvable hardware, as will be seen in the experiments). However, these properties can make the evolution converge to a sub-optimal solution. For numerous engineering problems, especially to evolve hardware with security, a sub-optimal solution will be an acceptable solution whenever that solution improves the performance of the system.

V. RESULTS FROM SIMULATIONS

The algorithm (μ,1+λ)-E1E was tested on the Ackley cost function:

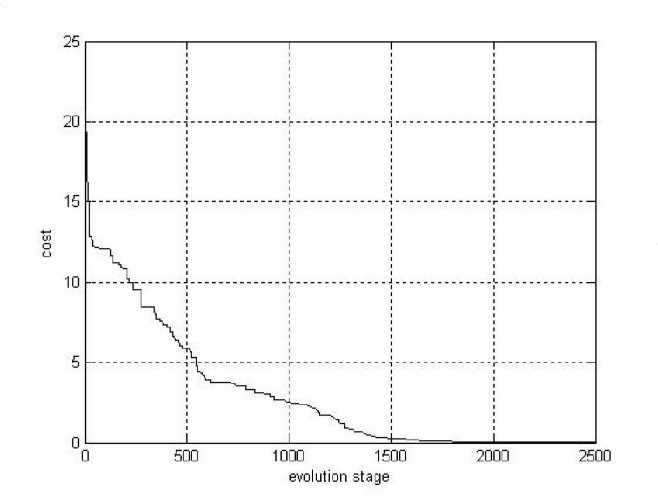

Figure 6. Cost of the fittest individual per evolution stage obtained from the test of (m,1+l)-E1E on the Ackley cost function.

$$F_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right) + 20 + e$$

where $x \in S$ is the characteristics vector of a individual; x_i is the i-th element of the vector x; and N is the dimension of space S. These functions have the global minimum at the origin. Space S is a real vector space with norm ∞ and 30 dimensions. This is a Banach space. The initial vector a_1^0 is such that $\|a_1^0\|_\infty \leq 30$.

Before testing the algorithm (μ,1+λ)-E1E, some heuristics are given.

A. Some heuristics for algorithm (m,1+l)-E1E

The self-adaptation of the dispersion works well when a large population is used. When a small population (for example $\mu=2$ and $\lambda=4$) and $c_n = \sqrt[N]{P_{cn}}$ with P_{cn}=0.5 are used, the dispersion tends to decrease all the time. This causes that the algorithm (μ,1+λ)-E1E does not converge to a minimum. To avoid this problem, with P_{cn}<0.5 should be used, for example P_{cn}=0.25. The probability to increase the dispersion is larger when a smaller P_{cn} is used but excessively small values of P_{cn} affect the precision of the results.

To increase the probability of avoiding local minimums, small P_{cn} should be used. For example P_{cn}=0.16.

To decrease the risk of breaking down a physical system in an evolvable hardware, a small value of r should be used. However, the value of r should not be excessively small because the convergence of the algorithm could be very slow. To increase the probability of avoiding local minimums a large value of r should be used.

The choice of the population of the species depends on the difficulty of the problem. More difficult problems need more population.

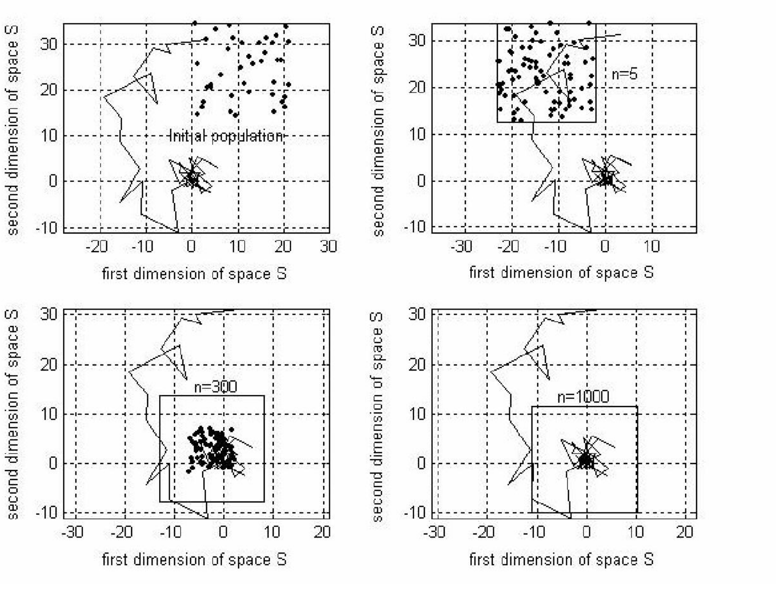

Figure 7. Trajectory generated on the first and second dimension of space S by the fittest individual. Results obtained from the test of (m,1+l)-E1E on the Ackley cost function. An individual is represented by a point. The closed ball $\tilde{B}(v'^{n-1};r/c_n)$ which contains the children at the n-th evolution stage is represented by a square.

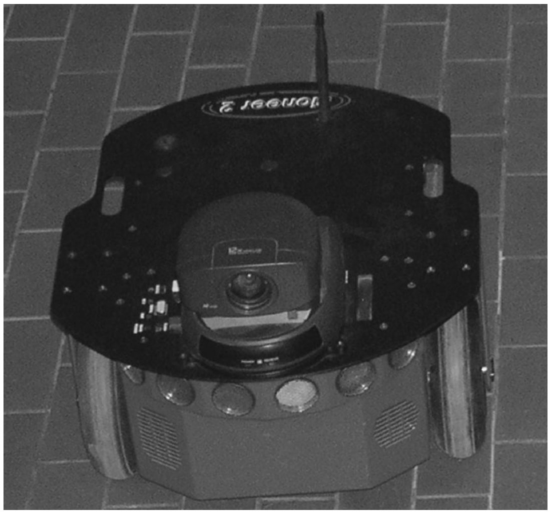

Figure 8. Mobile robot Pioneer 2DX

The relation of λ and μ can be λ=2μ , with this equality a good performance is obtained in the dispersion's self-adaptation.

B. Test of (m,1+l)-E1E on the Ackley cost function

The algorithm parameters are: r=10, μ=40, λ=80, $c_n = \sqrt[N]{P_{cn}}$ with P_{cn}=0.16. The result of the cost variation is shown in Fig. 6. The variation of the species dispersion and the trajectory generated by the fittest individual are shown, on the first and second dimension of space S, in Fig. 7. The number of function evaluations was 200000. The final cost, obtained after applying the algorithm (μ,1+λ)-E1E, is 5.5x10^{-3}. Additionally, ten independent tests were performed, in all of these tests the algorithm located the global minimum; the average final cost is 10.496 x10^{-3}. In [1] the same test was performed with ES and EP; results were: final cost 7.48x10^{-8} with μ=30, λ=200 for ES; and final cost 1.39x10^{-2} with μ=200 for EP.

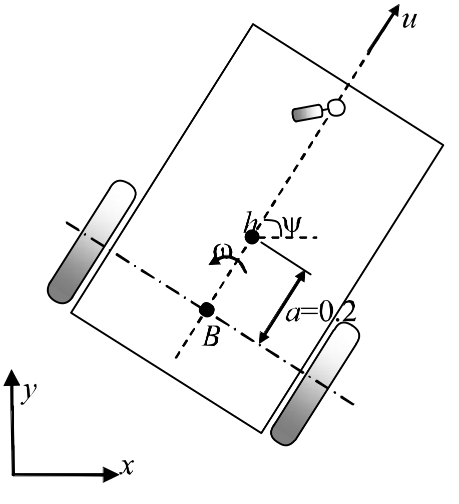

Figure 9. Schematic of the mobile robot

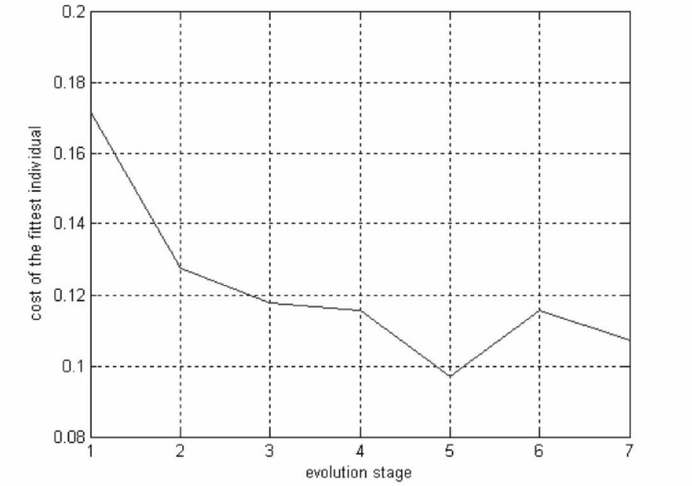

Figure 10. Experimental result: cost of the fittest individual per evolution stage

VI. EXPERIMENT

Let us consider as an individual a mobile robot (see Fig. 8) with a tracking control. The mobile robot is a unicycle like vehicle which has two traction wheels and one castor wheel (see Fig. 9). The point

$$h = [x \;\; y]^T$$

is the point that is required to track a trajectory. The trajectory tracking control of the robot system is obtained from [5]. This tracking control considers the dynamic model of the mobile robot in its control law to compensate the non-linear part of the system. The tracking control is presented as follows:

$$\nu_{ref} = \hat{D}M(\upsilon - N) + T_a\hat{\theta}$$

$$\upsilon = \ddot{h}_d + K_1\dot{\tilde{h}} + K_2\tilde{h}, \qquad \tilde{h} = h_d - h\,,$$

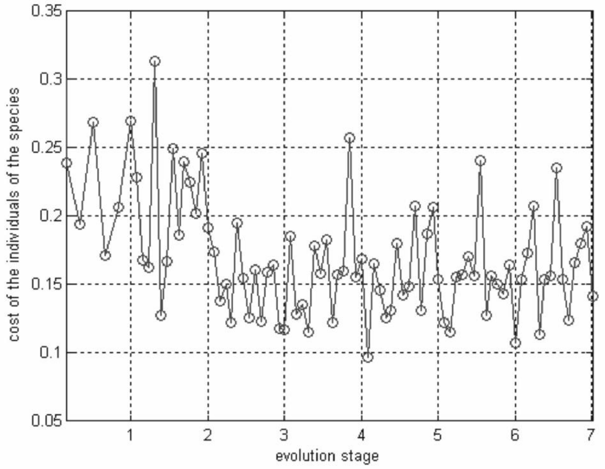

Figure 11. Experimental result: cost of all individuals of the species per evolution stage

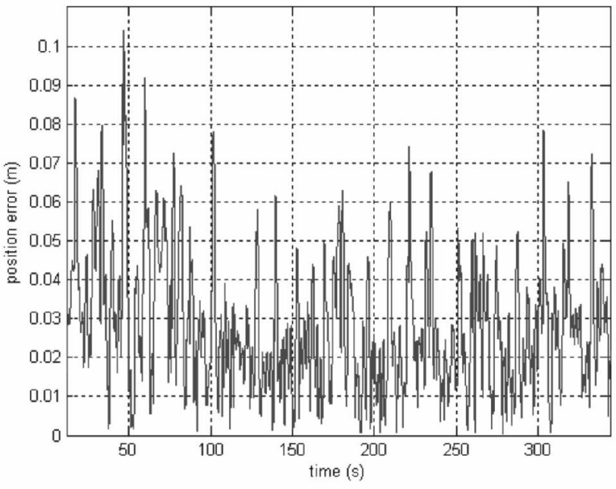

Figure 12. Position error of the tracking control during evolution

where:

$$\nu_{ref} = \begin{bmatrix} u_{ref} \\ \omega_{ref} \end{bmatrix}, \hat{D} = \begin{bmatrix} \hat{\theta}_1 & 0 \\ 0 & \hat{\theta}_2 \end{bmatrix}, M = \begin{bmatrix} \cos\psi & \sin\psi \\ -\frac{1}{a}\sin\psi & \frac{1}{a}\cos\psi \end{bmatrix},$$

$$N = \begin{bmatrix} -u\omega\sin\psi - a\omega^2\cos\psi \\ u\omega\cos\psi - a\omega^2\sin\psi \end{bmatrix},$$

$$T_a = \begin{bmatrix} 0 & 0 & -\omega^2 & u & 0 & 0 \\ 0 & 0 & 0 & 0 & u\omega & \omega \end{bmatrix}, \hat{\theta} = \begin{bmatrix} \hat{\theta}_1 & \hat{\theta}_2 & \dots & \hat{\theta}_6 \end{bmatrix}^{\mathrm{T}},$$

$$K_1 = \begin{bmatrix} 2\sqrt{0.5} & 0 \\ 0 & 2\sqrt{0.5} \end{bmatrix}, K_2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix},$$

$h_d(t)$ defines the desired trajectory, u and ω are the linear and angular velocity, u_{ref} and ω_{ref} are the linear and angular reference velocity (these references are send to the robot), $\hat{\theta}_i$ is the i-th estimated parameter of the dynamic model, and the rest of variables are shown in Fig. 9.

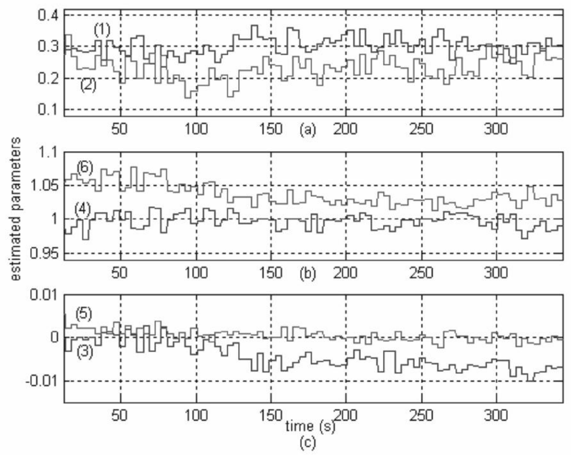

Figure 13. Estimated parameters of the dynamic model during evolution

The elements of the characteristics vector of the individual are the estimated parameters of the dynamic model of the robot. These estimated parameters are used to compensate the nonlinearities in the control system. Better estimated parameters imply better control performance. The problem here can be formulated as to obtain the estimated parameters that give the least control errors.

In the experiment, the cost of an individual is calculated by evaluating the performance of the individual (mobile robot) executing a task. The task considered in this experiment is that the robot must follow a circular trajectory of 0.5m of radius with a velocity of 0.4m/s. The cost function that evaluates the performance of the tracking control is:

$$F(\hat{\theta}) = \int_{t_1}^{t_2} \sqrt{\left(\tilde{h}^{\mathrm{T}}\tilde{h} + \dot{\tilde{h}}^{\mathrm{T}}\dot{\tilde{h}}\right)} dt$$

where t_1 and t_2 are the initial and final time instant of the evaluation of one individual. In the experiment $t_2 - t_1 = 4$ seconds. Figure 14 shows an example of the trajectory that the vehicle describes during the evaluation procedure.

The initial model parameters of the mobile robot, obtained by an identification method, are [4]:

$$\hat{\theta} = \begin{bmatrix} 0.3037 & 0.2768 & -0.0004 & 0.9835 & 0.0038 & 1.0725 \end{bmatrix}^{\mathrm{T}}.$$

One can observe that some parameters have greater values than others. Therefore, a normalized vector of parameters is considered as an individual. The space S is $\mathbf{R}^6$ with ∞-norm. The parameters of the evolutionary algorithm $(\mu, 1+\lambda)$-E1E are:

r=0.2, μ=6, λ=12, $c_n = \sqrt{P_{cn}}$, P_{cn}=0.5.

The value of r was chosen using the experiences that the authors have in this kind of tracking control.

In the experiment 7 evolution stages were performed. In each evolution stage, except the first, 13 evaluations were computed that correspond to 12 children and the last fittest individual, which is evaluated again. The first evolution stage evaluates only the initial population that is 6 individuals. Every evolution stage, except the first, lasts 52 seconds and every evaluation lasts 4 seconds. In order to wait that the system stabilizes, the first evaluation begins after 11.9 seconds since the control is initiated.

The result of the fittest individual cost variation per evolution stage obtained in the experiment is shown in Fig. 10. In this

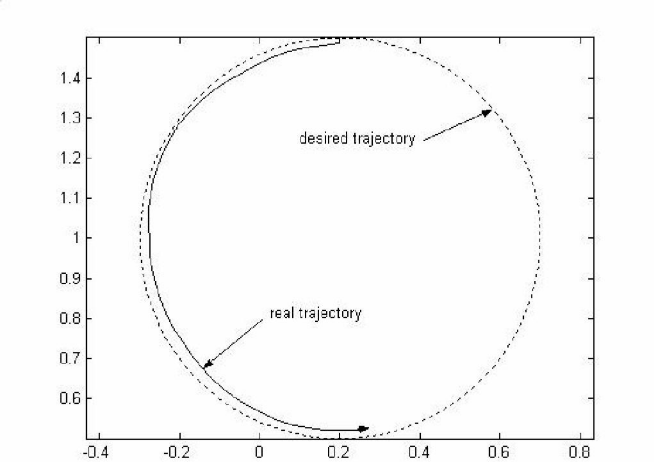

Figure 14. Experimental result: Evolution of the vehicle trajectory when the initial individual is evaluated.

figure at the 6-th evolution stage, one can see an increment of the cost of the fittest individual. This increment is caused by random perturbations in the cost function F. The function F is perturbed because the task that the individual executes depends on the initial condition, noise in the velocity measurements, variations of the processing time in the operative system (Windows), irregularities in the ground, etc. Thus, the task never is exactly repeated. Therefore, when the fittest individual is evaluated again, the obtained cost is a little different to the previous result. In Fig. 11 is shown the cost of all individuals of the species per evolution stage. One can see that the worst cost per evolution stage tends to decrease. This result demonstrates that the physical system works better and better during the evolution. Figure 12 shows the position error of the tracking control during the evolution. After 100s the control error are smaller than before 100s; that is the consequence of the optimization of the tracking control when using the proposed evolutionary algorithm. In Fig. 13 is presented the variation of the estimated parameters in time. A tendency to some value can be observed specially in 3-th and 6-th parameters. Figures 14, 15 and 16 show comparisons of trajectories that the vehicle describes when some individuals are evaluated. One can see that the trajectories in the last evolution stage perform better (see figures 15 and 16) than the obtained using the initial individual (see Fig. 14).

It is important in the experiment that function F must only depends on the estimated parameters. This is difficult to reach at, because F depends on the initial position, the random uncertainties, and other factors. However, good results can be obtained if F varies slowly through time and with small amplitude.

Remark: This experiment was performed only to demonstrate the functionality of the evolutionary algorithm (μ,1+λ)-E1E in a real situation. Better results (faster adaptation and less control errors) can be obtained using adaptive control and stability theories [5]. However, these theories can not be applied when a simple and exact enough model of the real system is not available. In contrast, to apply the evolutionary algorithm presented in this work, only an approximated model of the real system is required.

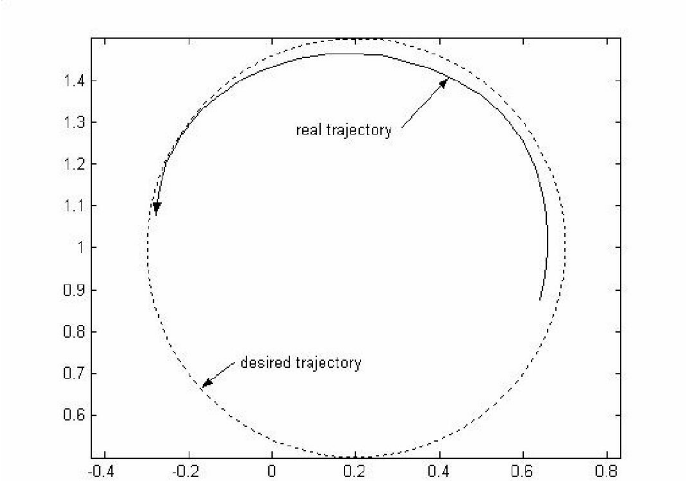

Figure 15. Experimental result: Evolution of the vehicle trajectory when the worst individual of the last evolution stage is evaluated.

VII. CONCLUSIÓN

The evolutionary algorithm (μ,1+λ)-E1E differs from the other algorithms developed so far on the fact that it shows good properties for hardware evolution. This is so because it can be guaranteed that the algorithm will not lead the system, when evaluating the costs function, to instability or to undesirable stances. Another property of the evolutionary algorithm (μ,1+λ)-E1E is that the self-adaptation of the species dispersion concentrates such species around the solution for tuning up the results. The convergence towards a minimum for the evolutionary algorithm (μ,1+λ)-E1E was proven by simulations. The experimental results validate the theoretical aspects of the evolutionary algorithm. Besides, the experiment shows that a real system can evolve without any risk of work badly while it is executing a repetitive task. It is important that the system executes a repetitive task because function F(.) must depend only on the characteristics vector of an individual.

Future work will analyze evolving hardware with function F(.) perturbed by random uncertainties. Besides, future work will analyze the use of multiple species in order to obtain a better search of the global minimum.

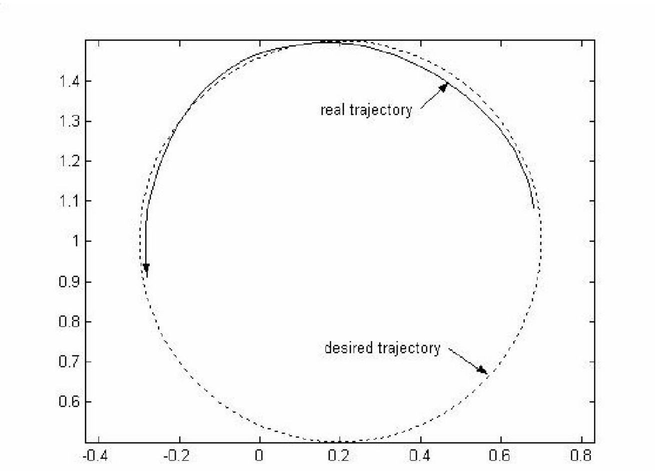

Figure 16. Experimental result: Evolution of the vehicle trajectory when the fittest individual of the last evolution stage is evaluated.

ACKNOWLEDGMENT

The authors would like to thanks to FAPES-Brazil for financing the project: Adaptive Dynamic Controller Applied to a Robotic Wheelchair Commanded by Brain Signals, process 39385183/2007. This work was also supported by the National Council for Scientific and Technical Research (CONICET), the National Agency for Scientific and Technological Promotion (ANPCyT) under grants PICT/04 No. 21592, PICT/05 Start-up No. 35398, and National University of San Juan under grants 21/I 842-2007.

REFERENCES

[1]T. Bäck, G. R. and SchwefelH.P., 1993. Evolutionary programming and evolution strategies: Similarities and differences. In: Fogel, Atmar, Eds, Second Annual Conference on Evolutionary Programming, La Jolla, California, pp. 11-22.

[2]Cai, Z., Zeng, S., Yang, Y. and Kang, L., 2008. Automated antenna design using normalized steady state genetic algorithm. NASA/ESA Conference on Adaptive Hardware and Systems, pp. 125-132.

[3]Darwin, C., 1995. The Origin of Species, Gramercy.

[4]De La Cruz, C. and Carelli, R., 2008. Dynamic model based formation control and obstacle avoidance of multi-robot systems. In: Robotica, Carelli, Vol. 26, pp. 345-356.

[5]De La Cruz, C., Carelli, R., and Bastos, T.F., 2008. Switching adaptive control of mobile robots. In: IEEE International Symposium on Industrial Electronics – ISIE08, Cambridge, UK.

[6]Eiben, A.E. and Smith, J.E., 2003. Introduction to Evolutionary Computing, Berlin Heidelberg, Springer-Verlag.

[7]Hornby, G.S., Takamura, S., Yamamoto, T. and Fujita, M., 2005. Autonomous evolution of dynamic gaits with two quadruped robots. In: IEEE Transactions on Robotics, Vol. 21, No 3, pp. 402-410.

[8]Kreyszig, E., 1978. Introductory Functional Analysis with Applications, New York, John Wiley & Sons.

[9]Lipson, H. and Pollack, J.B., 2000. Automatic design and manufacture of robotic lifeforms. In: Nature, Vol. 406, pp. 974-978.

[10]Lohn, J.D. and Hornby, G.S., 2006. Evolvable hardware: using evolutionary computation to design and optimize hardware systems. In: IEEE Computational Intelligence, Vol. 1, No. 1, pp. 19-27.

[11]Mingzhi, X., Weicai, Z. and Licheng, J., 2003. The convergence of the abstract evolutionary algorithm based on a special selection mechanism. In: IEEE International Conference on Computational Intelligence and Multimedia Applications, pp. 356-361.

[12]Thompson, A., 1997. An evolved circuit, intrinsic in silicon, entwined with physics. In: 1st International Conference on Evolvable Systems 1996, Springer Verlag.

Celso De La Cruz received his Ph.D. in Control Systems Engineering from "Universidad Nacional de San Juan", Argentina, in 2006. He is currently a Post-Doctoral Student in Electrical Engineering at "Universidade Federal do Espírito Santo", Brazil. His current research interests include evolutionary computation analysis, dynamic modeling and control design of mobile vehicles.

Teodiano Freire Bastos graduated in Electrical Engineering at Universidade Federal do Espirito Santo - UFES (Vitoria, Brazil), in 1987. He received his Specialization in Automation in 1989 at Instituto de Automatica Industrial (Madrid, Spain) and his Ph.D. in 1994 in Physical Science (Electricity and Electronics) at Universidad Complutense de Madrid (Spain). He is currently an Associate Professor at UFES. His current research interests are mobile robots and assistive technology for handicapped people.

H. Daniel Patiño received the electronic engineering degree, and the doctorate degree in engineering from the Universidad Nacional de San Juan (UNSJ), Argentina, in 1986 and 1995 respectively. He is an Associate Professor in the Instituto de Automática and the Electronics and Automatics Department, Faculty of Engineering, UNSJ. His current research areas cover computational intelligence, advanced optimal control, approximate adaptive dynamic programming, robotics, systems and control theory, data mining, and computational finance.

Ricardo Carelli obtained a Ph.D degree in Electrical Engineering from the National University of Mexico (UNAM). He is full professor at the National University of San Juan and senior researcher of the National Council for Scientific and Technical Research (CONICET, Argentina). Prof. Carelli is Director of the Instituto de Automática, National University of San Juan (Argentina). His research interests are on Robotics, Manufacturing Systems, Adaptive Control and Artificial Intelligence Applied to Automatic Control.