An Evolutionary Approach to Designing Autonomous Planetary Rovers

Martin Penaik*, Barry Bentley*, Davide Marrocco*,
Christos Ampatzis†, Dario Izzo† & Francesco Biscani†
*Centre for Robotics and Neural Systems, University of Plymouth
†Advanced Concepts Team, European Space Agency

***Abstract*— Current methods used to address the problem of autonomous navigation in planetary rovers rely on computationally expensive algorithms and elaborate 3D sensing strategies. This paper presents a low complexity alternative based on an evolved neural controller using continuous-value infrared sensors. A unified framework is presented for developing such controllers in virtual planetary rovers, utilising the latest advances in parallel optimisation by way of island model evolution. Preliminary results are presented demonstrating that this approach is capable of producing successful navigation and obstacle avoidance behaviours, showing that equivalent or better results can be achieved relative to traditional genetic algorithms.**

I. Introduction

An imperative problem currently faced in the robotic exploration of the Solar System pertains to the fundamental communication limits imposed by the vast distances of space, limits which make the real-time remote control of space robots impossible. As a result, the continued future success of unmanned space exploration relies heavily on the ability to design robust autonomous systems.

A. Background

The Mars Pathfinder mission, launched by NASA in 1997, was the first to successfully land a semi-autonomous vehicle on the Martian surface (named Sojourner), a success which was later repeated and surpassed in 2004 with the dual-landing of the Mars Exploration Rovers (MER) Spirit and Opportunity. While designed to function for only 90 days in the harsh Martian environment, both Spirit and Opportunity are continuing to perform experiments and transmit important scientific data after half a decade of near-continuous operation [1]. There are still however many unanswered questions regarding Mars' atmosphere, geology and possible biology for which more sophisticated instruments are required to answer. Many of these answers will be essential in paving the way for future crewed missions to the planet, a goal set out by the US government on 14th January 2004 in *The Vision for Space Exploration*. In order to address these questions, both NASA and the European Space Agency have scheduled further robotic missions to the planet, namely the Mars Science Laboratory (MSL) and ExoMars missions respectively.

As discussed above, the ability to act autonomously is a necessary prerequisite for any space probe. In the case of a planetary rover, the most essential autonomous capabilities required to prevent a mission failure are those of efficient navigation and obstacle avoidance within unknown environments. In a complex planetary terrain, there are many environmental factors that need to be considered for this, such as gradient, roughness, rocks, holes and terrain stability (a factor which recently contributed to the permanent immobilisation of the Spirit rover [1]).

To facilitate navigation and obstacle avoidance behaviours, all of the successful rovers to date (i.e. Sojourner, Spirit and Opportunity) have utilised stereo-camera sensing. In the case of the two most recent vehicles — Spirit and Opportunity — three sets of stereo-camera pairs are installed: one fore-facing, one aft-facing, with a third primary pair situated on the mast. The data from these cameras is processed using a combination of stereo algorithms to construct a 3D representation of the immediate terrain as well as a local *traversability map* [2]. These are then used to select the next action of the robot.

Other well-studied methods for navigating unknown environments include:

- *The arcs approach:* In this method a 3D model of the environment is produced upon which several candidate arcs are generated. These arcs are then evaluated according to some utility function and the best one is selected to steer the robot [7], [8].
- *Behaviour-based navigation:* Here, obstacle avoidance behaviours emerge from a complex interaction of elementary sub-behaviours which are predefined by a programmer [5], [6].
- *Grid-type traversability maps:* The steering of the robot is calculated by mapping the immediate environment into a grid, which is then searched to find the most efficient route [9].

B. Objectives & rationale

Given the above-listed methods currently employed in the navigation of unknown environments, the au-

thors sought to develop a framework to investigate the potential for less algorithmically-complex control system utilising alternative fault-tolerant sensing methods — objectives especially pertinent to the domain of space exploration, where computational resources and human intervention are both severely constrained [4].

As discussed, all of the rovers currently deployed depend upon 3D cameras for navigation — an inherent mission vulnerability. Should these cameras fail, it follows that the robots would be left incapable of safely navigating their environment in an autonomous manner. For this reason, it is worth exploring complimentary sensing systems.

C. Approach

A unified framework and test platform were developed to explore the potential of controllers based upon the principles of *evolutionary robotics* (ER). In this paradigm the control system of a robot comprises of an artificial neural network (ANN) whose synaptic weights are optimised by way of the Darwinian principle of *selective reproduction with variation* [3]. The rationale for choosing ANNs as the control system stems from their low computational overhead, generalisation abilities, and tolerance to input noise.

Recent attempts in evolutionary robotics to deal with the problem of traversing rough terrains have focused largely on the idea of coordinated motion behaviour, where several interconnected mini-robots act collaboratively to produce complex locomotive strategies [10]. Other strategies have looked at the use of morphologically-reconfigurable robots capable of adapting to the physical properties of their environment [11], [12], [13].

In contrast to the studies mentioned above, this work uses a 3D physics simulation of a single robot modelled on the MSL rover, deployed in a Mars-like environment and implemented using the Open Dynamics Engine (ODE), an open source library for simulating rigid body dynamics (http://www.ode.org). To date, two minimal-complexity control architectures have been studied on this simulation platform: 1) a simple feed-forward neural controller using binary infrared sensors with evolvable activation thresholds [14], and 2) a recurrent neural-network based active-vision system [15].

The study detailed herein looks to build upon the original work mentioned in point one above by adopting floating-point infrared sensors for increased terrain sensitivity, and a control system optimised using a parallel island-model genetic algorithm (GA) — an evolutionary strategy originally developed by Cohoon et al. [19], [20] and based upon the theory of *punctuated equilibria* [16]. This theory postulates that speciation in the natural world arises from brief periods of rapid evolution punctuated by long periods of evolutionary stasis (see [17], [18]), of the sort that might occur in archipelagoes where the populations in separate islands diverge over time, undergoing rapid evolution when new solutions enter the population via inter-island migration.

Applying the island paradigm to artificial evolution affords the ability to split a large population into several subpopulations, or islands, which can be evolved in parallel. This has the effect of drastically reducing the evolution time. To facilitate the sharing of genetic material between these subpopulations, the best individuals from each island are selected at predefined intervals and exchanged to simulate the process of inter-island migration. This has the additional benefit of preventing any one population converging to a local optima, as shown by Ampatzis et al. [22] in a first application of the paradigm to the optimisation of neural controllers. The island model paradigm has also proven successful in the parallelisation of other global optimisation algorithms when applied to difficult and high dimensional problems [21]. It is the aim of the authors to extend this success of the island model in evolutionary robotics, and in particular to the domain of planetary rover navigation by integrating the Mars rover physics simulator with the Parallel Global Multiobjective Optimiser (PaGMO), an island model framework developed by the European Space Agency's Advanced Concepts Team (http://sourceforge.net/projects/pagmo).

In the study described in the following sections, the neural controllers were evolved using the island model, with each island evaluating against an environment featuring a combination of rocks and holes. In this study is has been found that the island model is capable of evolving solutions of comparable quality to traditional, sequential GAs, while significantly reducing the time required to do so. The method in which this was achieved is delineated in the following section.

II. Method

As outlined in the introduction, the approach adopted in this work is that of evolutionary robotics (ER), an emerging paradigm that draws upon Darwinian principles to exploit the important coupling between an embodied agent and its environment [23], resulting in extremely powerful sensory-motor abilities [24]. To date, the complexity of neuro-controller solutions derived from the evolutionary method is lower than systems designed with expert knowledge. To address this shortcoming, the island model is being applied to the problem of evolving neural controllers, which has resulted not only in significant time reductions in the optimisation process, but has also been found capable of producing fitter individuals [22]. The nature of island demes, allows for a much wider

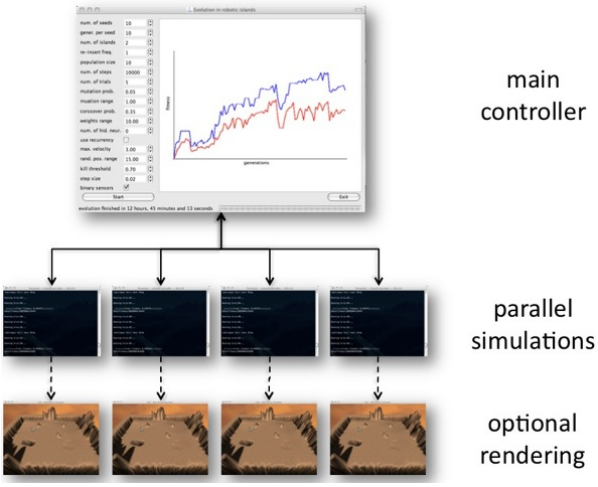

Fig. 1. Architectural diagram showing the relationship between the controller and physics simulator.

exploration of the search space, while the introduction of new individuals from other islands via migration prevents any one of these populations converging on a local optima. An added advantage of this model is the ability to study more complex ER scenarios, such as evolving separate populations for different tasks in different islands, using migration to integrate these behaviours.

The exact set up and applications that were explored in this study are detailed below.

A. Simulator

The simulator consists of two logically separate parts: a controller and a physics simulation of the rover and its environment (see Fig.1 & 2).

The controller incorporates the functionality of island evolution through the PaGMO libraries as well as handling the configuration of simulation parameters, such as the neural network architecture, terrain selection, sensor configuration, graphics and environmental properties.

The physics simulator executes the evaluation of neural controllers (i.e. genotypes) by deploying them in the environment and returning a floating-point number to the controller representing the achieved fitness of that controller, as defined by a fitness function (explained below). Due to the logical independence of the simulator and the controller, it is possible to launch several simulations concurrently, one for each island. As the Open Dynamics Engine does not necessitate rendering the simulation, this model is in no way dependant upon graphics.

B. Rover Model

The robot used in this experiment is a 3D physics simulation model of the MSL rover. The model cannot be considered an accurate or detailed representation of the actual rover, but only an approximation. This is primarily due to the paucity of information published on the rover's dimensions, mass distribution, component properties, as well as many other details. According to the Centre National d'Etudes Spatiales [25], the dimensions of the real rover are 2900 mm $\times$ 2700 mm $\times$ 2200 mm, with a total mass of approximately 775 kg. The physics model of the rover was therefore built using these these details and modelled on the several diagrams available. This modelling imprecision is not crucial to this study however, as the aim is not to construct an accurate simulation of the MSL, but rather demonstrate the application of evolutionary robotics in developing a suitable controller for planetary rovers.

The motor system of the rover model (see Fig.3) consists of six wheels where the two front and the two rear wheels are able turn up to 90° to either side. Through the use of a rocker-bogie suspension system the rover is capable of surmounting obstacles that are of the same size as its wheels. This advanced suspension system is designed to operate at low speeds and consists of two pivoted joints connecting two bogies with two rockers [26]. These rockers are connected together via a differential join, meaning the left and right parts of the rocker-bogie system can move independently while keeping the rover body parallel to the ground.

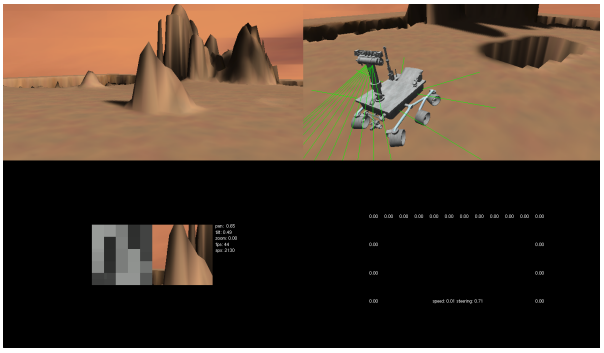

Fig. 2. Physics simulation of the Mars rover. The right section shows the user-controlled camera, the rover and the sensor inputs. The left section shows the rover's field of vision and information from the active vision system when in use.

The sensory apparatus used in the rover for this study consists of 18 infrared sensors, which are used to provide information about objects in the immediate proximity of the rover. Two different sets of sensors are used to accommodate obstacle detection. The first of these sets consists of six lateral sensors, affording safety when approaching obstacles from the side. These sensors have a range of three meters and cover an area of approximately 200° around the rover, leaving the front area deliberately exposed. Previous work on the rover implemented a binary sensor model, returning 1 when an obstacle was present, and 0 when nothing was in range [14]. This study has looked to

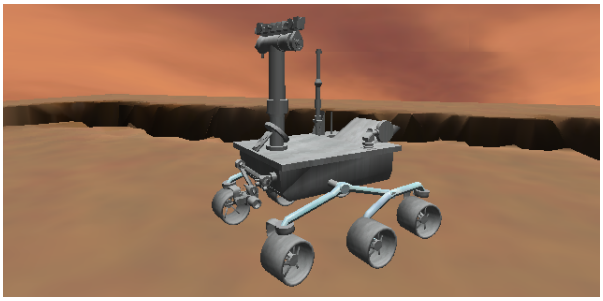

Fig. 3. 3D physics model of the rover showing the different parts of the rocker-bogie suspension system.

expand the fidelity of these sensors and as such, continuous value, floating-point sensors have now been implemented, returning a value between 0 and 1, with the value depending on the distance of the object from the rover. As before, 0 indicates no sensory contact, however 1 now indicates physical contact with the rover — all values in-between correlate to the object's distance.

The second set consists of 12 infrared sensors with a maximum range of five and half meters. These infrared sensors, which shall be referred to as ground sensors, are positioned on the rover's camera mast and point downward at a 45° angle, reaching the ground approximately three meters in front of the rover. The twelve sensors are positioned and directed to ensure the range extends to around 400 mm beyond ground level. Ground sensors constantly scan the distance from the surface and are able to detect both rocks and holes. As with the lateral sensors, each of the ground sensors returns a floating-point value from 0 (no feedback) to 1 (strongest feedback). Holes or cliffs can be detected by the rover when it loses sensory feedback from the ground (i.e. a ground sensor returns 0). The same sensors also allow the robot to detect dangerous rocks or excessively rough terrain. Previously this was achieved through an evolvable threshold, which would return 1 if either the sensor value exceeded the threshold (indicating a rock or dangerously rough terrain), or if there was no contact with the sensor (indicating a hole) — this way, a 1 would be returned whenever the rover was required to take action.

In this latest study, one of the aims was to see if this same behaviour could be produced without the use of a threshold, passing a floating-point value in the range $[0, 1]$ directly to the neural network; again the reason being to increase sensor fidelity and also to test how the island paradigm deals with this additional problem complexity. To increase the sensitivity at the most active part of the sensor's range, an effective length was defined, ranging from the end of the sensor's reach, to half of the beam's length — all values above this are returned as 1 (see Fig.4). To clarify, any object that intersects the sensor beam at $\leq 50\%$ total length, results in a 1 being returned. Holes return a 0, and all values in-between are represented with a floating-point number between 0 and 1. In addition to the sensors mentioned, the rover is equipped with an active vision system (not used in the experiments reported here) and two internal sensors measuring its speed and steering angle.

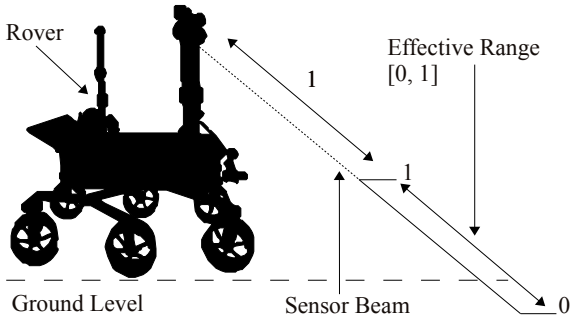

Fig. 4. Effective sensor range.

C. System Architecture

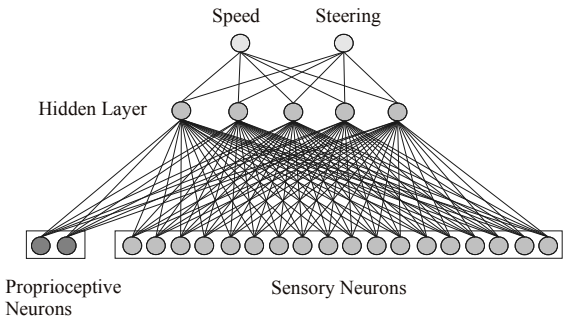

Fig. 5. Feed-forward neural network used as a control system for the rover in the evolutionary experiments.

The control system of the rover is a fully-connected, discrete-time, feed-forward ANN (See Fig.5). It consists of 18 exteroceptive neurons, each activated by one the rover's 18 infrared sensors, with 2 additional proprioceptive neurons encoding the values returned by the internal-state sensors (wheel orientation and speed). The 20 sensory neurons are fully connected to a hidden layer of 5 neurons, which is in turn fully connected to 2 motor neurons that modulate the level of force applied to the actuators — responsible for the rover's speed and steering. These motor neurons have the sigmoidal activation function

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

in the range $[0, 1]$, where x is the weighted sum of the inputs minus the bias. Biases are implemented as weights on the hidden and output neurons with

an activation value set to -1. The rover's actions depend on the values of the synaptic weights of the ANN, ergo each weight must be set to an appropriate value to produce a desired output and, as mentioned previously, a genetic algorithm is used to evolve these. The parameters that constitute the genotype of the control system and that are subject to evolution consist of 117 genes — the 100 synaptic weights that connect the 20 sensory neurons to the 5 hidden neurons and the 10 synaptic weights that connect the hidden layer to the motor neurons, plus the 7 hidden and output layer biases. Weights and biases are encoded as floating point values in the range [-10, 10].

D. Evolution

The island model is based on virtual archipelagoes, defined as chains or clusters of islands (subpoupulations) with specific migration routes between them. In this study an archipelago was constructed consisting of 9 islands, each with 10 randomly initialised individuals (every island having a different seed), giving a total population of 90 individuals, all of which were evaluated in the same environment containing 2 rocks and 2 holes (Fig.6 top). Within the PaGMO library, feasible

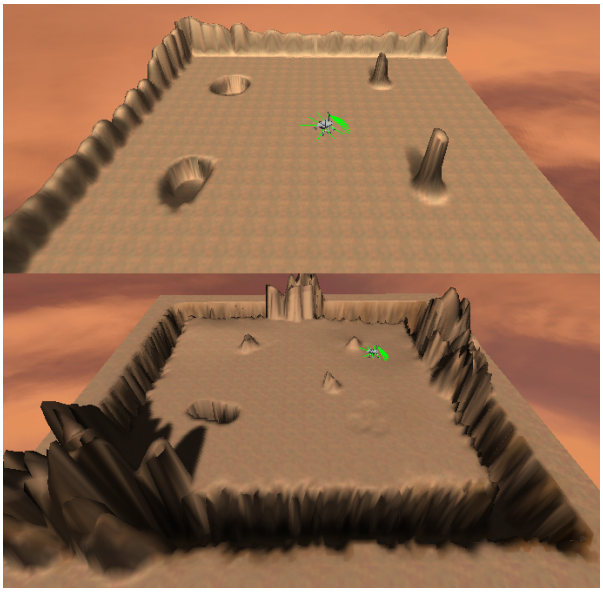

Fig. 6. Top: Environment used during evolution. Bottom: Environment used for testing the final individuals.

migration paths are given by particular topologies, examples of which include chain, ring, cartwheel, ladder, hypercube, lattice and broadcast topologies. For this experiment the ring topology (Fig.7) was utilised with migration commencing every 5th generation. All of the islands were evolved concurrently using a genetic algorithm, each island having a population size of 10 individuals with the best 2 individuals producing 5 offspring at each generation. Mutation

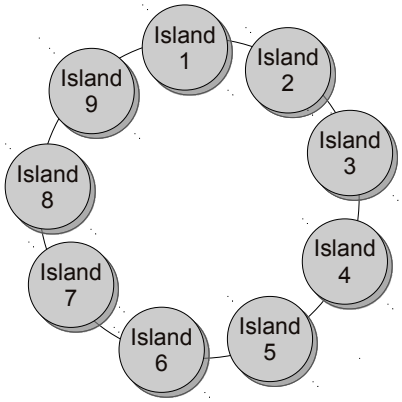

Fig. 7. Ring migration topology connecting individual islands and defining the rule sets for the exchange of genetic material between them.

was subsequently applied to these offspring with a 5% probability of adding a value to the original gene in the range $[-1, 1]$. The best individual of the previous generation was retained unchanged, replacing the worst of the 10 offspring (known as elitism). During each generation, all of the genotypes were evaluated 10 times for 3000 sensory-motor cycles (i.e. 3000 activations of the ANN), each time initialising the rover with a different starting position and orientation. This whole process was repeated for 100 generations, with 10 replications being conducted in total, starting each time with a different set of randomly generated individuals distributed across the 9 islands.

The performance of each control system was evaluated according to the fitness function in eq. 2, that was carefully designed to shape the behaviour of the robot for effective, reliable exploration and obstacle avoidance behaviours

$$F = \frac{1}{S \cdot T}(Sp \cdot St) \tag{2}$$

where the fitness F is a function of the measured speed Sp and steering angle St, where Sp and St are in the range [0,1]. Speed Sp is 1 when the rover is at maximum speed and 0 when it is stationary or reversing. Steering angle St is 1 when wheels are straight and 0 when they are turned over an angle of $30\,^{\circ}$ from the centre. For example, if the steering angle was $15\,^{\circ}$ then St would be 0.5. T is the number of trials (10 in these experiments) and S is the number of sensory-motor cycles per trial (3000 in these experiments). Equation 2 shows how the fitness is evaluated at every sensory-motor cycle. Thus, the GA has to maximise the fitness by increasing the value of Sp and St, which implies that a rover has to move at the maximum possible speed while steering only when necessary. If a rover goes forward at the maximum speed but consistently steers at an angle over $30\,^{\circ}$ then its final fitness will be 0. Similarly, if a rover travels backwards or sits

idle, its fitness will also be 0 regardless of the steering angle. The maximum fitness contribution at each time step is therefore $1/(S \cdot T)$. The final fitness of each individual is in the range [0, 1] and it is the average of all contributions from all time steps of all trials.

To test the final solutions, a more complex environment was used (see Fig.6 top), featuring inclined and declined surfaces, three high and three small rocks, rough areas and holes. 111 m^2 of the terrain was covered by obstacles and hence not traversable. Both of the environments used were 60 m $\times$ 60 m in area.

III. Results

An experiment was conducted using the above set up to test the ability of the island model to produce suitable controllers for planetary rovers. As an experimental control, a set of optimisations were carried out using the traditional, single population approach. To ensure parity, both approaches used the same evolution parameters and evaluated the same number of individuals (90 individuals $\times$ 100 generations). It was found that both approaches produced controllers capable of navigating in unknown environments, avoiding obstacles of different types. 10 replications were performed

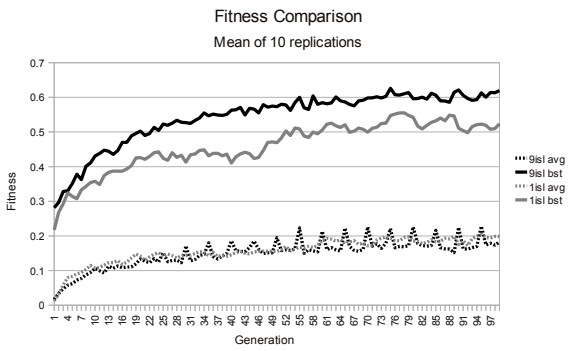

Fig. 8. Average fitness for each generation from all ten replications. The island model is plotted in black, the standard approach is plotted in grey. Solid lines represent the mean of the maximum fitness acheived across all replications. Dashed lines represent average fitness across all replications

for both approaches — the averages of all replications can be seen in Fig.8, showing that the island model consistently achieves higher maximum fitness across all generations (solid black line). The main advantage to the island model, besides from achieving higher maximum fitness, lies in its ability to parallelise the evolutionary process, with a time decrease directional proportional to the number of processors employed. In this case, the population was split over 9 islands with each island running on a separate core, effectively reducing the total optimisation time by a factor of 9. This result empirically proves that the evolution of neural controllers can be parallelised without degrading the results in any way.

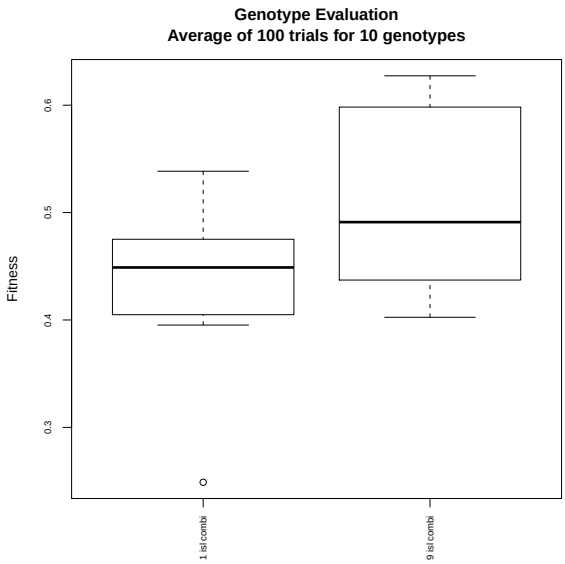

Fig. 9. Evaluation of final genotypes.

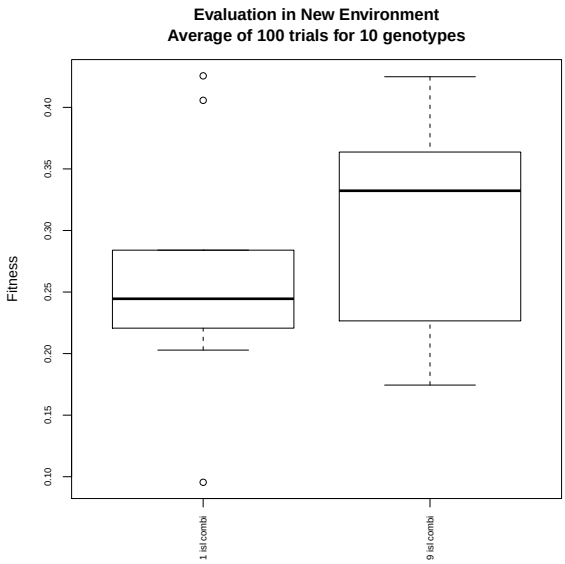

Fig. 10. Evaluation of final genotypes in unseen environment.

The behaviour of the controllers are consistent with what was expected from the fitness function — rovers travel in a straight trajectory, steering only when it is necessary to avoid obstacles.

To test the robustness of the solutions, the final individual from each replication was evaluated for an additional 100 trials in the evolution environment (see Fig.9) as well as a new environment, previously unseen by the controller (see Fig.10). As is shown in

the plots, the island model solutions on average outperform those from the single population, and perform much better relative to the traditional approach when presented with a new and more complex environment.

IV. Conclusions

It has been empirically demonstrated that the island model can produce neural controllers for planetary rovers, capable of navigating and avoiding obstacles in unknown environments, while using a continuous value sensor system. It has been shown that this approach can achieve equivalent or better results than the classical evolutionary approach, while drastically reducing the time required for a solution.

This preliminary demonstration has shown the potential of the island model in evolutionary robotics, and more specifically, the application of new design strategies and alternative sensing methods to the design of planetary rovers. As these alternative approaches continue to progress, it is posited that they might well provide solutions to problems that are currently intractable using conventional methods.

Acknowledgments

The authors gratefully acknowledge support from the ARIADNA scheme of The European Space Agency. Computational experiments were conducted using Apple's Xserver suite as part of an Apple ARTS award to Davide Marocco and Angelo Cangelosi.

References

[1] JPL, (2010, February) Summary. [Online]. Available: http://marsrovers.jpl.nasa.gov/overview/

[2] S. Goldberg, M. Maimone, and L. Matthies, "Stereo vision and rover navigation software for planetary exploration," in *IEEE Aerospace Conference*, Big Sky, Montana, March 2002.

[3] S. Nolfi and D. Floreano, *Evolutionary Robotics*. Cambridge, Massachusetts: MIT Press, 2001.

[4] M. Maimone, A. Johnson, Y. Cheng, R. Willson, and L. Matthies, "Autonomous navigation results from the mars exploration rover (mer) mission," in *Proceedings of the 9th International Symposium on Experimental Robotics (ISER)*, Singapore, June 2004.

[5] D. Langer and et al., "A behavior-based system for off-road navigation," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 776–783, 1994.

[6] H. Seraji and A. Howard, "Behaviour-based robot navigation on challenging terrain: a fuzzy logic approach," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, 2002.

[7] S. Singh and et al., "Recent progress in local and global traversability for planetary rovers," in *IEEE International Conference on Robotics and Automation*, 2000, pp. 1194–1200.

[8] S. Lacroix and et al., "Autonomous rover navigation on unknown terrain: Functions and integration," *International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 917–942, 2002.

[9] C. Ye and J. Borestein, "A method for mobile robot navigation on rough terrain," in *IEEE International Conference on Robotics and Automation*, 2004, pp. 3863–3869.

[10] V. Trianni, S. Nolfi, and M. Dorigo, "Hole avoidance: Experiments in coordinated motion on rough terrain," in *Proceedings of IAS-8: Intelligent Autonomous Systems Conference*, 2004, pp. 29–36.

[11] A. Castano, W. Shen, and P. Will, "Conro: Towards deployable robots with inter-robot metamorphic capabilities," *Autonomous Robots*, vol. 8, pp. 309–324, 2000.

[12] K. Sty, W. M. Shen, and P. Will, "Global locomotion from local interaction in self-reconfigurable robots," in *Proceedings of the 7th International Conference on Intelligent Autonomous Systems (IAS-7)*, W. M. Shen, C. Torras, and H. Yuasa, Eds. Amsterdam, The Netherlands: IOS Press, 2002.

[13] M. Yim, D. G. Du, and K. D. Roufas, "Polybot: A modular reconfigurable robot," in *Proceedings of the 2000 IEEE/RAS International Conference on Robotics and Automation*, vol. 1, 2000, pp. 514–520.

[14] M. Peniak, D.Marocco, and A. Cangelosi, "Co-evolving controller and sensing abilities in a simulated mars rover explorer," in *IEEE Congress on Evolutionary Computation (CEC)*, Trondheim Norway, May 2009.

[15] M. Peniak, D. Marocco, S. Ramirez-Contla, and A. Cangelosi, "An active vision system for navigating unknown environments: An evolutionary robotics approach for space research," in *Proceedings of IJCAI-09 Workshop on Artificial Intelligence in Space*, Pasadena, California, July 2009.

[16] N.Eldredge and S. J. Gould, *Punctuated Equilibria: An Alternative to Phyletic Gradualism*. Freeman, Cooper and Co.New York, NY, 1972.

[17] S. Bornholdt and K. Sneppen, "Neutral mutations and punctuated equilibrium in evolving genetic networks," *Physical Review Letters*, vol. 81, no. 1, pp. 236–239, 1998.

[18] P. Bak and S. Boettcher, "Self-organized criticality and punctuated equilibria," *Physica D: Nonlinear Phenomena*, vol. 107, no. 2-4, pp. 143–150, 1997.

[19] J. Cohoon, S. Hegde, W. Martin, and D. Richards, "Punctuated equilibria: a parallel genetic algorithm," in *Proceedings of the Second International Conference on Genetic Algorithms and their application*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1987, pp. 148–154.

[20] W. N. Martin, J. Lienig, and J. P. Cohoon, *Island (migration) models: evolutionary algorithms based on punctuated equilibria*. Institute of Physics Publishing, Bristol, UK, 1997, ch. C6.3:1-C6.3:16.

[21] D. Izzo, M. Rucinski, and C. Ampatzis, "Parallel global optimisation meta-heuristics using an asynchronous island-model," in *Proceedings of the IEEE Conference on Evolutionary Computation (CEC 2009)*, 2009.

[22] C. Ampatzis, D. Izzo, M. Rucinski, and F. Biscani, "Alife in the galapagos: migration effects on neuro-controller design," in *Proceedings of the European Conference on Artificial Life (ECAL 2009)*, 2009.

[23] R. Beer, "A dynamical systems perspective on agent environment interaction," *Artificial Intelligence*, vol. 72, pp. 173–215, 1995.

[24] S. Nolfi, "Power and limits of reactive agents," *Neurocomputing*, vol. 42, no. 119-145, 2002.

[25] CNES. (2010, February) Msl at a glance. [Online]. Available: http://www.cnes.fr/web/5719-msl-09-at-a-glance.php

[26] D. P. Miller and T. L. Lee, "High-speed traversal of rough terrain using a rocker-bogie mobility system," in *Proceedings of Robotics 2002: The 5th International Conference and Exposition on Robotics for Challenging Situations and Environments*, 2002.