# Co-Evolution of Sensor Morphology and Control on a Simulated Legged Robot

Gary B. Parker (*Member, IEEE*) and Pramod J. Nathan

*Abstract*— **This paper discusses utilizing genetic algorithms to automatically design a suitable sensor morphology and controller for a given task in categories of environments. The type of sensors, the heading angle and the range of the sensor, and the rules the controller, are co-evolved. The described method enables the system to decipher information from the environment to determine that is relevant to completing a given task while configuring a minimal controller and number of sensors, thus increasing the overall efficiency of the robot.**

## I. Introduction

AUTOMATIC design is a fairly recent and exciting option in the search for new and efficient physical and software designs for autonomous agents. Self designing and optimizing agents offers a new design perspective as it allows each agent to adjust its design according to its inherent advantages/disadvantages and adapt its features to achieve a goal. The work presented here is a continuation of research pertaining to the evolution of sensor morphology on a legged robot. This study deals with evolving the controller and the sensor morphology on a legged robot. We choose to evolve sensors since they play a key role in deciphering the environment in which an agent performs. To be successful, an agent must adjust its sensors to pick out information relevant to its task and learn to ignore the others while also being energy efficient. This is an important feature seen in nature: different species have a different set of sensory organs that vary in the degree of their sensitivity and other parameters. No animal has every sensory organ since such an agent would not be efficient. Moreover, certain tasks performed by animals need only a subset of the animal's available sensory organs and capabilities. Likewise, autonomous agents need only a small subset of their available capabilities for specialized tasks. Utilizing a subset of the available capabilities for a given task further allows the agent to be more energy efficient and decreases the complexity of the controller.

Researchers have applied the idea of evolutionary morphology to evolve the positioning of an array of the same type of sensors [1] and to evolve the body and sensors of the robot simultaneously [2]. However, these works evolved sensor morphology using the same type of sensors. The drawback to this is that sensors of the same type have limited detectable stimuli, thus reducing the capabilities of the robot as a whole. Other research used modular parts to build structures [3] and robots for the purpose of locomotion; this work on buildable structures shows us that the idea of evolving morphologies is a feasible and practical solution for design problems.

The field of co-evolution of morphology and the controller has been increasingly researched in recent years. Research done on Tinkerbots [4] used the idea of re-usable sub-procedures called Lindenmayer systems to design scalable complex designs. Similarly, Marbach and Ijspeert [5] present an implementation of co-evolution of a PD controller and morphology using simulated modules to build scalable simulated mobile robots. Yet as pointed out by Pollack *et al.*, [6] these agents do not interact with the environment due to the lack of sensors that allow it to sense the environment. This absence of sensors does not allow these agents to evolve beyond a certain complexity due to the lack of constant feedback from the environment. Balakrishnan and Honavar [7] evolved the position and ranges of sensors but in a highly limited and discrete simulated block environment. Research conducted by Bugajska and Schultz [8] and Mautner and Belew [9] shows strategies to co-evolve the robot controller and the sensor morphology. Unlike our study, the former study dealt with finding general sensor morphology for any environment configuration, whereas in the latter the complexity of the environment was constant. Our study deals with specific environment configurations of varied complexity. LEGO Mindstorms robots were used by Lund *et al.* [10] in co-evolving the controller and the physical and sensor morphology of the robot. But the search space was limited (825 possibilities) with only a single type of sensor being used with 11 possible positions; there is no such limitation in our study. The work of Bugajska and Schultz [8] did not utilize different types of sensors or utilize stimuli other than obstacles. In addition their robot has an onboard sensor that can detect distance to goal. Our robot does not have this; instead, the agent must learn to find the goal using cues available in the environment.

There is a relationship between the morphology, the controller, and the environment of a given system [11] [12]. For a specific task certain stimuli provided by the environment are advantageous and others are not. An agent that is capable of sensing these stimuli needs to determine which stimuli are needed to complete the task at hand and prune the unnecessary ones, making itself more efficient in terms of power consumption and controller complexity

Manuscript received 31 Jan 2007.
G.B. Parker is the Director of Computer Science at Connecticut College, New London, CT, USA (e-mail: parker@conncoll.edu)
P.J. Nathan is a graduate from Connecticut College, New London, CT, USA (e-mail: pjnat@conncoll.edu)

(hence improving reaction time). In previous work, we successfully evolved the sensor morphology of a legged robot to complete a navigation task which involves going from a start point to a target point. The agent avoided obstacles and utilized relevant stimuli in the environment to navigate to the target point. However, the agent in that work always had the same heading and position at the start. This allowed the agent to learn an appropriate sequence of actions for the environment, making the solution specific to a starting position and heading in the specific environment. In the research reported in this paper we wanted the robot to have a sensor morphology and controller that could not rely on its starting position or orientation to succeed. We show that by co-evolving sensor morphology and control we can design autonomous agents that can complete a given task efficiently using stimuli in the environment as cues, while having a level of generality in terms of its start position and orientation. To investigate this method we use a simulation of the robot operating in a real world environment.

In this paper we use a genetic algorithm (GA) to co-evolve the sensor morphology and controller for a simulated robot modeled after an actual legged robot. This is part of a larger study that investigates the use of a GA to evolve the morphology and a controller for actual robots. In this part of the study, we evolve the sensor morphology and controller to equip the robot to complete a given navigation task when it starts from a randomized start position and heading, in a wide variety of environments.

## II. Robot Colony and ServoBot

The robot colony space is a 3m x 3m walled area established in the Connecticut College Robotics Laboratory. It can be equipped with robots and can be configured with any reasonable number of obstacles and light sources. The obstacles are 30cm x 30cm x 5cm blocks. The lights are two omnidirectional light sources of different frequencies. The obstacles are not high enough to block the light beam. Both the obstacles and the walls can be detected by the tactile sensors placed on the robot.

The ServoBot [13] is a hexapod robot that has 2 degrees of freedom per leg. We have equipped it with a sensor base that is attached to the top of the robot and completely covers the robot so that the actuators and effectors of the robot do not affect the sensors. This base is 30cm x 30cm and acts as an easily reconfigurable platform for the sensors. Unlike wheeled robots, which can turn at any angle less than or equal to its maximum turn angle, the legged robot with specific gaits does not have this ability. It has 16 available turn gaits, each of which has a predicted (through measurement) end orientation and position given a start orientation and position. The agent's locomotion is non-deterministic, i.e. given a start position and orientation, the end position and orientation after a step has a degree of randomness involved. This non-determinism in the gait is due to the inconsistencies of the robot construction and inherent noise associated with a legged robot interacting with the real world.

The sensor base can carry modules of 4 tactile sensors and 8 light sensors, 4 of which detect Infrared and 4 that detect Ultraviolet which is interfaced with a Basic Stamp II. Using a Basic Stamp II along with the sensor modules restricts the number of sensors used to 12. The sensors are all binary; they either detect a stimulus or do not. Each of the sensors is a reusable module that can interface with the controller. The GA is responsible for evolving the orientation of the sensors and the range of the light sensors. The range at which the light sensors detect a stimulus is implemented in software by setting a cutoff at which the controller no longer detects a stimulus. The GA has 32 equal length choices for the range of each light sensor. In simulation the maximum range of these sensors is set at 434 cm. This range was chosen since it is slightly more than the hypotenuse of the Robot Colony which is approximately 424 cm. The spread of the sensors was not evolved since no mechanism was in place to adjust this aspect of the sensor. The tactile sensors were placed in the corners of the sensor base and the light sensors were placed on the edges of the sensor base.

## III. Simulation: the Environment and Agent

The colony space was represented by a 300 X 300 area in simulation. The agent's task was to traverse from Start Position (40 +/- 10, 150 +/-10) to the Target Position (270, 160). The agent itself has no information about the location of the target point. To navigate it must learn and use cues from the environment to its advantage.

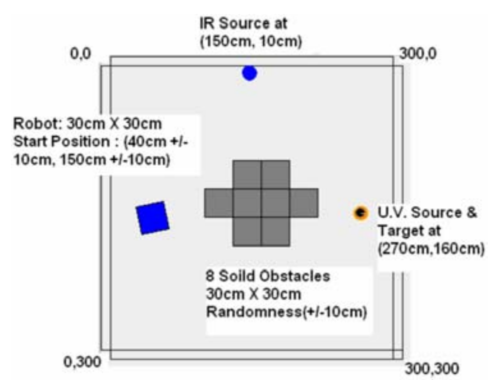



Fig. 1. Environment Description (Central Mountain Configuration)

The Niche Environments (Figure 1) used eight 30 x 30 obstacles and two omnidirectional light sources (IR and UV). In previous experiments a sound source was simulated but due to high noise levels caused by reflection of sound, it was discarded in favor of the more controllable light stimuli. The robot could detect light over the obstacles and the tactile sensors detected walls and obstacles. The obstacles were configured in 7 different configurations (Figure 5). For each generation the obstacles were moved randomly by a factor equal to or less than +/- 10cm in the X and the Y direction to allow the results to have a level of generality. The obstacle density was 8%. The target was a fixed point marked by one of the light sources.

The simulated robot closely models the ServoBot. Its locomotion characteristics were obtained by measuring the performance of the actual robot. It is simulated to have a 30cm x 30cm reconfigurable platform for the sensors. Its locomotion is non-deterministic in that the measured turn values are used but randomized slightly.

The sensor base can be configured with 4 tactile sensors, 4 infrared sensors and 4 ultraviolet sensors. The light sensors have adjustable range (32 length choices), whereas the tactile lengths are fixed. The placement, maximum range and spread of the sensors on the sensor base are shown as triangles in Figure 2. (The IR sensor and the UV Sensor range and spread overlap). The light sensors are placed facing outward, with the tactile sensors placed in the corners of the sensor base. The GA determined which sensors would be activated and their range (for light) and orientation. The evolved characteristics allowed the sensory information to be complex enough for the robot to be successful in the environment while making the simplifications necessary to allow ease of transfer to the real robot.

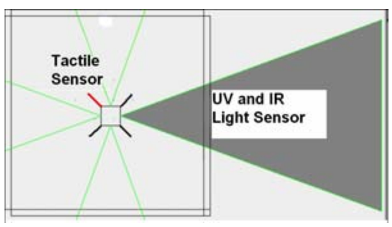



Fig. 2. Sensor Configuration (Range, Spread and Placement of Sensors on Sensor Base

The controller of the simulated robot is a reactive system that used 13 rules of the form: If (Sensor detects a stimulus) then (Trigger gait number X), in which each sensor is associated with a specific rule and a single gait. The consequents of these 13 rules are selected by the GA from the 16 available gaits [13]. These turn gaits used were from data stored after measuring the 16 gaits on the actual robot. The GA also selects the default gait, which is the gait used when no sensors are triggered (rule 13).

## IV. Co-Evolution of Sensor Morphology and Control

The parameters evolved in this paper are the heading angle of the sensors, which sensors to keep running during the length of the run, the range of the light sensors, and the rules of the robot controller. The sensor heading can be rotated 360 degrees; all of the sensors can be turned on or off. The GA can select the controller for each sensor from 16 turn gaits, while each light sensor has 32 choices of range. The position of the sensors on the base are fixed as shown in Figure 2 and do not change.

The chromosome used was 212 bits long. It was divided into 1 set of 12 bits, 12 sets of 9 bits, 12 sets of 5 bits and 13 sets of 4 bits. The first 12 bits represented which sensors to put on or off, the next 108 bits represent the angles at which each of the 12 sensors are to be placed onboard the robot base, 40 bits represent the range of the light sensors and the last 52 bits represent gaits for each of the 13 rules. During evolution each set of bits underwent a single point crossover using stochastic (roulette wheel) selection of the parents. A mutation rate of 1% was set for the rule selection. For the other parameters, if the goal was reached by any individual in the generation the mutation rate was set at 1%; otherwise it was set at 6%.

The fitness function (Figure 3) is dependant on the success of the agent. If the agent is successful in finding the target, the fitness is based on the number of sensors it has off and the amount of time it took the agent to get to the target. The fitness of an unsuccessful agent is dependent on how far away the robot is from the goal. In the first case, to achieve the maximum fitness the robot has to have all its sensors off and reach the goal without any time being used, which is an impossible scenario in the simulation. A successful agent has the theoretical maximum fitness of 15600 while an unsuccessful agent has a maximum fitness of only 8480.

We ran 256 individuals for 512 generations for each environment. This was repeated 5 times with random starting populations to check for consistency of the results. Each individual had 3 chances to achieve its goal. Each time the individual was placed at a random heading and positioned within +/- 10 of (40,150). The individual with the highest fitness was added to the next generation without change.

The agent was given 300 time intervals to complete the task. A time interval is the time it takes a robot to complete one step. A step starts with the legs in a ready to step position (right front and back legs and left middle leg forward; remaining legs back) and returns to this position after a full step cycle. This time is the same for all 16 step cycles. The time intervals continue to elapse even if the robot collides with an obstacle and is stuck. A collision does not stop a test run. The agent can get out of a collision since the agent can work its way free of the obstacle due to its non-deterministic gait. The run is stopped if the target position is reached or when the agent runs out of time (300 time intervals).

```
If (Agent_Reached_Goal)
  Fitness =
    50 * Number_Sensors_Turned_Off +
    50 * (Total_Time - Time_to_Achieve_Goal) +
    Goal_Bonus
Else
  Fitness = 20 * Resultant_Distance_from_Goal
```

Fig. 3. The Fitness Function

## V. Results and Discussion

The results of the 5 runs on each environment are summarized in Figure 4. These graphs show the average and best individual of the population, over 5 runs, for each generation. The consistency of the results over the 5 runs is represented by the error bars (grey) in the graphs which are the standard deviation over the runs.

In all of the niche environments, except for Random Two, the GA evolved effective solutions and improved the fitness of the agents. Each of the environments had a different growth rate and became stable after a different number of generations. This shows that success and control learning is highly dependent on the environment in which the agent operates. In Random Two, the agents failed to find the Target point over 512 generations. The fitness of the average individual in this case is approximately 27.29%, while the average fitness of the elites is 43.6%. In the other environments the final obtained fitness of the average and the elites obtained is markedly higher than that of Random Two.

Figure 5 is a sample from each of the environment types. Observations of the simulated robots in action revealed some of the learned strategies. The evolved strategies were a combination of wall-following and a more direct approach to finding the source of the UV light, hence the target point. Wherever there was a clear path to the Target Point the GA evolved agents that took the direct approach. This can be seen in Double Ridge and Random Three. In both cases, the main sensor is the UV light sensor which is mounted heading forward. Random Three also evolved a tactile sensor on its left flank to take into consideration any change in position of obstacles to its left over the generations.

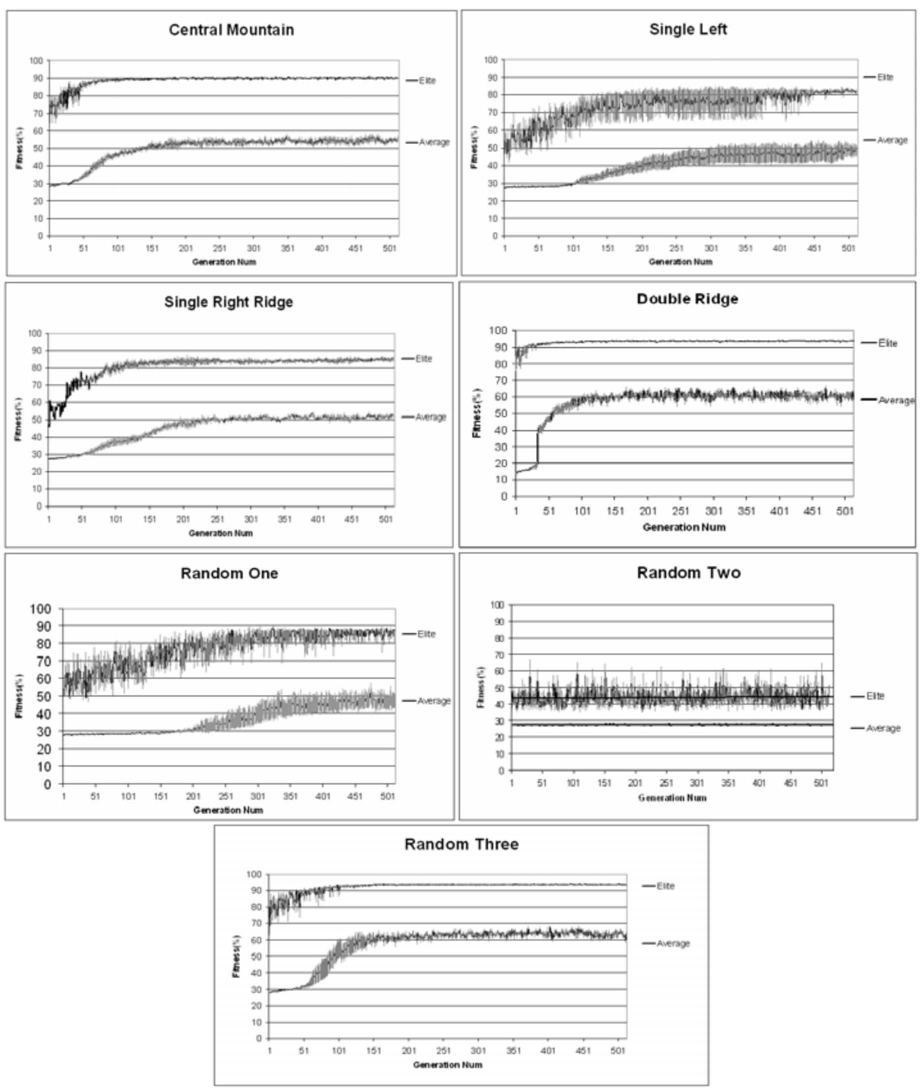
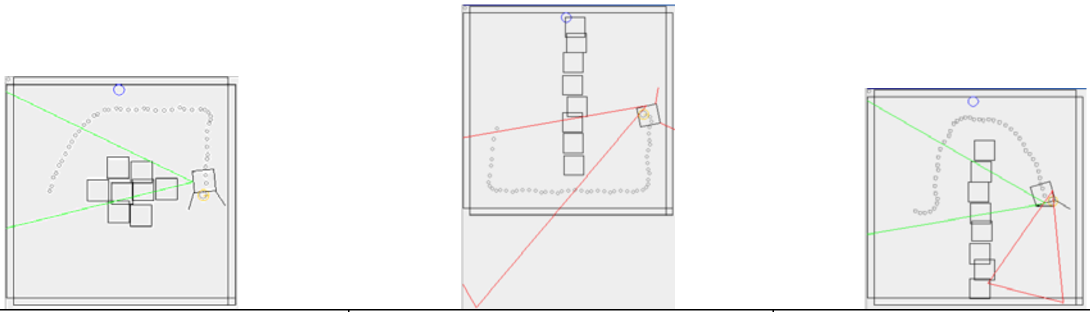
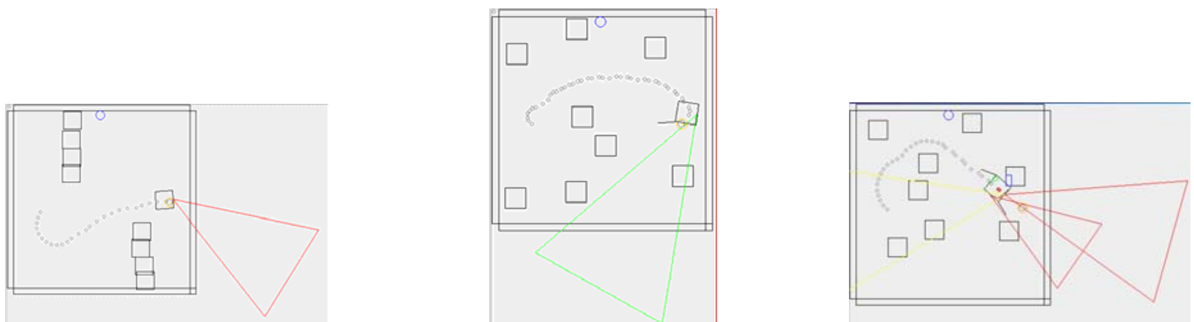


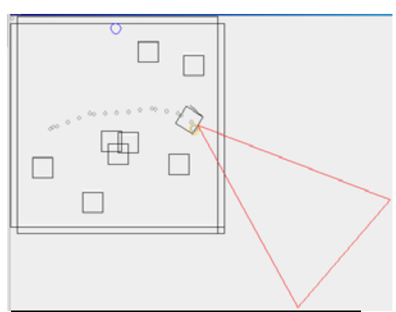

Fig. 4. Elite and Average Fitness growth averaged over 5 Runs



| Central Mountain: | Single Left Ridge: | Single Right Ridge: |
|---|---|---|
| 2 Tactile sensors at 22 and 333<br>1 UV sensor at 101; Length 378 cm<br>4 Gaits and Rules Used | 2 Tactile sensors at 381 and 128 UV sensor at 252; Length 406cm<br>4 Gaits and Rules Used | 2 Tactile sensors at 287 and 319<br>2 UV sensors at 117 and 32; Length 378cm, 204cm<br>5 Gaits and Rules Used |



| Double Ridge: | Random One: | Random Two (UNSUCCESSFUL): |
|---|---|---|
| 1 Light Sensor Used at 221; Length 140cm<br>2 Gaits and Rules Used | 1 Tactile Sensor at 78<br>1 UV sensor at 21<br>3 Gaits and Rules Used | 3 Tactile sensors at 18, 457, 157<br>1IR sensor at 487; Length 392cm<br>3 UV sensors at 353, 334, 282; Length 182cm, 294cm, 14cm<br>8 Gaits and Rules Used |



| Random Three: |
|---|
| 1 Tactile Sensor at 187<br>1 UV sensor at 9; Length 308 cm<br>3 Gaits and Rules Used |

Fig. 5. Paths Taken and Sensors Used by the Robot

The agents in Central Mountain, Single Left Ridge and Single Right Ridge used a combination of wall following and UV source location. In all three cases the agent used the UV sensor to direct it initially in an appropriate heading. In Central Mountain and Single Left Ridge, the agent then used a wall following strategy to find the Target Point. In Single Right Ridge, the agent used the tactile sensor only to get out of the narrow gap between the wall and the obstacle. It also evolved a tactile sensor pointing inwards so that if it's right turn out of the gap was too close to the obstacle it could detect it and turn away. This is important since the solid obstacles were moved for each generation. The agent then used its UV sensor to find the Target Source.

In the case of Random Two the agent could not consistently find the Target Point. This is apparent since the average fitness of the population does not increase nor does the fitness of the elites over a given run. The reason for this failure is the difficulty of this particular environment. The Random Two environment has a configuration of solid obstacles that does not often allow the agent to pass between its gaps. Where there are sufficient gaps, the randomness associated with the placement of the obstacles results in random location of the gaps. The inconsistency of this situation varied the fitness of the individuals sufficiently that there was little or no improvement in acquiring the target over time. This lack of improvement under such an inconsistent situation can be associated with the purely reactive nature of the controller which does not allow for planning of any kind.

It was surprising to find that none of the evolved agents used the IR light as a possible way point to finding its way to the UV light and hence the Target. The penalty of having a sensor on was only 2% of the total fitness, yet none of the successful agents utilized this stimulus. The IR source's placement was intended to be advantageous. The agent could have used the UV light to navigate around obstacles then get to the target in each environment, especially Single Left Ridge and Single Right Ridge. A closer look at the environment yields an answer to the agent's lack of IR sensor usage. The obstacles are low enough that light can be detected over them. A look at Single Right Ridge shows that the robot can circle in place until its right light sensor detects the target. At this point it will be facing north. It can track north until it detects a wall, moves east and eventually south east to turn away from the wall and will detect the target UV source with its front and slightly right facing sensors.

## VI. Conclusion and Future Work

The method used in this research to co-evolve the sensor morphology and controller for a hexapod robot was successful in simulation. Sensor morphologies and controllers were evolved that were specialized to handle types of environments whereas randomizing the robot's start position and orientation showed that the result is a generalized solution for a given environment. In all but one of the test environments the evolved robots improved throughout the evolution and produced reasonable sensor configurations with and associated rules.

This co-evolution strategy designed robots and their controllers that were successful in achieving the target and pruning the unnecessary sensors so that it was done efficiently. This can be clearly seen in Figure 4 since almost all of the successful agents in the latter half of the evolution were among 80% to 90% fit.

One of the more interesting parts of the results was that the infra red stimulus was not used at all to successfully navigate to the target point. It was intended that this information would be useful. However the GA developed solutions that did not require the extra sensor type. This design information could save significant time and money since these sensors would not have to be purchased or built for the robot. In this way this co-evolution strategy is beneficial before actual building as it allows pruning of unnecessary modules that would not improve the performance and make the robot more expensive than necessary.

The agents evolved were highly specialized in that they were not successful in the other environments. Their sensor configuration and control were extremely varied and unique for the different environments. This can be attributed to the reactive nature of the controller and a fitness function that rewarded efficiency in sensor usage. The GA evolved successful and in some cases familiar control strategies, like wall-following.

In future work, we will implement the results of this experiment on the actual ServoBot to allow us to see how accurately the simulation transfers to the real world. We will also conduct an experiment where the obstacles are tall enough to obscure the UV Light thus possibly making the IR light necessary for a reasonable solution.

In addition, we will consider a more complex controller architecture to co-evolve. The controller used consists of simple if…then rules, which allow us to measure the complexity of the controller with ease, but also severely limit the capabilities of the robot. We would like to make the controller more complex and test the results in a more challenging environment.

## References


[1] M. Fend, M. Yokoi, and R. Pfeifer, "Optimal Morphology of a Biologically-Inspired Whisker Array on an Obstacle-Avoiding Robot," Proceedings of the 7th European Conference on Artificial Life, ECAL Dortmund, Germany, 2003. .

[2] H.H. Lund , J. Hallam, and W. Lee, "Evolving Robot Morphology" IEEE International Conference on Evolutionary Computation, Indianapolis, USA1997.

[3] P. Funes and J. Pollack, "Computer Evolution of Buildable Objects" Proceedings of the Fourth European Conference on Artificial Life, Cambridge, USA, 1997.

[4] G. Hornby, H. Lipson, and J. Pollack, "Evolution of Generative Design Systems for Modular Physical Robots," Proceedings of the IEEE International Conference on Robotics and Automation, pages 4146–4151, 2001.

[5] D. Marbach. and A. J. Ijspeert, "Co-evolution of configuration and Control for Homogenous Modular Robots", In F. Groen *et al*., editor, Proceedings of the Eighth Conference on Intelligent Autonomous Systems (IAS8), pp.712-719. IOS Press, 2004.

[6] J. Pollack, H. Lipson, G. Hornby, and P. Funes, "Three Generations of Automatically Designed Robots," Artificial Life Vol. 7, Issue 3, pages 215 – 223, 2001.

[7] K. Balakrisha and V. Honavar, "On Sensor Evolution in Robotics," Proceedings of the First Annual Conference on Genetic Programming, Stanford University, USA, 1996.

[8] M. Bugajska, and A. Schultz, "Coevolution of Form and Function in the Design of Micro Air Vehicles," Proceedings of the NASA/DoD Conference on Evolvable Hardware, Alexandria, VA, USA, 2002.

[9] C. Mautner and R. Belew, "Evolving Robot Morphology and Control," Proceedings of Artificial Life and Robotics, Oita, 1999.

[10] H.H. Lund "Co-evolving Controller Morphology with LEGO robots" In Hara and Pfeifer (eds.) Morpho-functional Machines, Springer-Verlag, Heidelberg.

[11] R. Pfeifer, "Building 'Fungus Eaters': Design Principles of Autonomous Agents," Proceedings of the Forth International Conference on Simulation of Adaptive Behavior, Cambridge, USA, 1996.

[12] M. Atkins and P. Cohen, "Monitoring Strategies of Embedded Agents: Experiments and Analysis," Proceedings from Animals to Animats 4, Cambridge, USA, 1996.

[13] G. Parker, "Evolving Cyclic Control for a Hexapod Performing Area Coverage" IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2001.