View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by DigitalCommons@Connecticut College

Connecticut College
Digital Commons @ Connecticut College

Computer Science Faculty Publications | Computer Science Department

7-2010

Concurrently Evolving Sensor Morphology and Control for a Hexapod Robot

Gary Parker
Connecticut College, parker@conncoll.edu

Pramod J. Nathan

Follow this and additional works at: http://digitalcommons.conncoll.edu/comscifacpub

Part of the Computer Engineering Commons

Recommended Citation
Parker, G.B.; Nathan, P.J., "Concurrently evolving sensor morphology and control for a hexapod robot," Evolutionary Computation (CEC), 2010 IEEE Congress on , vol., no., pp.1,6, 18-23 July 2010 doi: 10.1109/CEC.2010.5586233

This Conference Proceeding is brought to you for free and open access by the Computer Science Department at Digital Commons @ Connecticut College. It has been accepted for inclusion in Computer Science Faculty Publications by an authorized administrator of Digital Commons @ Connecticut College. For more information, please contact bpancier@conncoll.edu.
The views expressed in this paper are solely those of the author.

Concurrently Evolving Sensor Morphology and Control for a Hexapod Robot

Comments

©2010 IEEE

DOI:10.1109/CEC.2010.5586233

This conference proceeding is available at Digital Commons @ Connecticut College: http://digitalcommons.conncoll.edu/comscifacpub/24

Concurrently Evolving Sensor Morphology and Control for a Hexapod Robot

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

***Abstract*— Evolving a robot's sensor morphology along with its control program has the potential to significantly improve its effectiveness in completing the assigned task, plus accommodates the possibility of allowing it to adapt to significant changes in the environment. In previous work, we presented a learning system where the angle, range, and type of sensors on a hexapod robot, along with the control program, were evolved. The evolution was done in simulation and the tests, which were also done in simulation, showed that effective sensor morphologies and control programs could be co-learned by the system. In this paper, we describe the learning system and show that the simulated results are confirmed by tests on the actual hexapod robot.**

I. Introduction

BOTH the control program and the morphology of a robot are important in the performance of its tasks and both are tied to the environment. Depending on the environment and the stimuli available, a learning system can find the most effective combination of sensors and control instructions to perform the given task. In this paper, the task is set – the robot is to navigate from its present location to a goal location. The types of sensors available are also set, with two types of light sensors and one type of tactile feeler sensor. Although the walls of the colony space and the positions of two light stimuli are also set, the locations of eight obstacles are placed with some level of randomness within the bounds of a predefined pattern. The learning that takes place is to determine the sensors that are needed, their placement, and the control program appropriate for each general category of environmental pattern.

Some interesting research has been done in the area of evolving morphology and control. Evolving sensor morphology has been applied to sets of the same type of sensor [1,2], optimization of a compound eye [3], and the simultaneous design of the controller and sensors of a robot [1,4,5]. Each of these studies uses only one type of stimulus from the environment to perform its task. Building on this work, this paper considers learning where the robot has three different stimuli that it can sense from the environment. Using a single stimulus as a key to the environment reduces the agent's capabilities and ability adapt to environmental change, especially if that stimulus can no longer be used or has a significant change.

Balakrishnan and Honavar [6] evolved the position and ranges of sensors in a limited and discrete simulated block environment. Bugajsaka and Schultz [7] showed strategies to find a general sensor morphology for any environment using obstacle detecting sensors and an onboard sensor to detect distance to goal. Mautner and Belew [4] used the co-evolution of the robot controller and the sensor morphology in an environment of constant complexity. Sugira et al. [8] used the co-evolution of sensors and a neural network controller to show that the environment affected the sensitivity and resolution of the sensors. In the research presented in this paper, we consider the ability of the learning system to find the sensors and their configurations required, and the control program needed for the robot operating in specific environment configurations of varied complexity. In addition to learning in an environment with a significant amount of noise, the robot in this study does not have a sensor that detects its distance to goal. Its only means of finding the goal is through stimuli that it senses from the environment.

In this paper, a method of evolving both the robot's sensor morphology and control [9] using a genetic algorithm (GA) is described, along with the results of tests done on the actual hexapod robot. The solutions learning in simulation are executed with the results favorably comparing to the simulation results.

II. The Environment

The environment was set up in a 3m x 3m walled colony space in the Connecticut College Robotics Laboratory. It can be equipped with robots and any reasonable number of obstacles and stimuli. Fig. 1 shows the colony space, two of its walls, the robot with its sensor platform, one of the light stimuli, and the eight obstacles placed in the Central Mountain configuration. The solid walls are high enough that the tactile sensors consider them to be the same as obstacles. The obstacles are 30cm x 30cm and low enough that the light stimuli can be seen over them. Eight of these obstacles, resulting in an 8% obstacle density, were used for the experiments described in this paper. They were placed in four distinct configurations. Two omni-directional light sources were placed on or near two of the walls of the colony. The infrared light was on the East wall and the ultraviolet light was placed near the South wall. The six

G. B. Parker is with the Computer Science Department at Connecticut College, New London, CT 06320 USA (corresponding author: 860-439-5208; e-mail: parker@conncoll.edu).

P. J. Nathan recently graduated with a Masters degree in Computer Science from Texas A&M University, College Station, TX 77843 USA (e-mail: pjnat@tamu.edu).

environmental configurations used for the actual robot tests were: Central Mountain, Single Left Ridge, Single Right Ridge, Double Ridge, and two that were randomly generated (Fig 6). For each test, the obstacles were placed in these configurations with their actual locations randomly moved by +/- 10cm in both the x and y coordinates. This presented the learning system with several distinct configurations to learn the best sensor morphology / control system for a general category of environmental configuration.

Fig. 1. Photograph of the colony space. The hexapod robot appears as a white square since mostly what one can see is the sensor base. Two tactile sensors are also clearly seen coming from the sensor base. The obstacles are 30cm x 30cm boxes; shown in the Central Mountain configuration. The light is a normal incandescent light since only one light sensor was used in each of the learned sensor morphology.

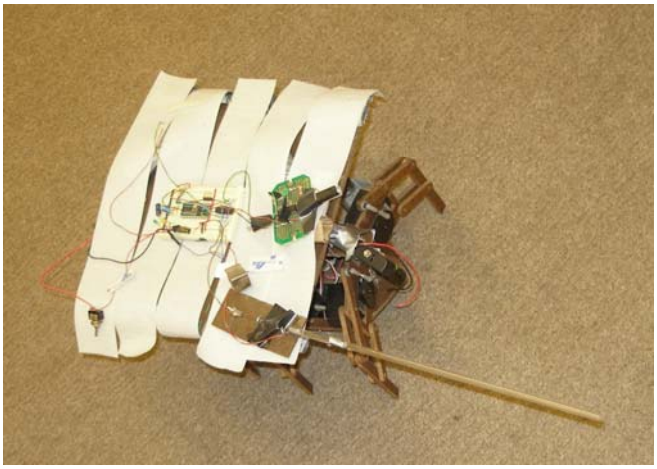


Fig. 2. Photograph of the ServoBot robot with sensor base configured with a light and tactile sensor.

A. The Robot

The robot used for these experiments was the ServoBot (Fig 2), which is a hexapod robot with 2 degrees of freedom per leg that are provided by 2 servomotors per leg. The robot walks using a gait initially generating using a cyclic genetic algorithm [10], but modified to produce 16 gaits that include 7 left turns, 7 right turns, a straight gait and a reverse gait. A Basic Stamp II controller is used to control the robot's locomotion by sending the appropriate signals to the individual servos to produce each of these gaits.

The ServoBot used for these experiments was equipped with a 30cm x 30cm plate that is attached to the top of the robot to serve as a sensor base. This sensor base, which completely covers the top of the robot, is covered with Velcro and serves as an easily reconfigurable platform for the sensors. It can carry modules of up to 4 tactile sensors, 4 infrared sensors, and 4 ultraviolet sensors. A Basic Stamp II is used to control all the sensors, which limits the total number of sensors to 12.

The light sensors have a range that is adjustable from 0 to 434cm in 14cm increments. The maximum range of these sensors is slightly more than the hypotenuse of the Robot Colony (424cm). The range of each tactile sensor (length of the feeler) is 25cm. An example of the placement, maximum range, and spread of the sensors on the sensor base is shown in Figure 3.

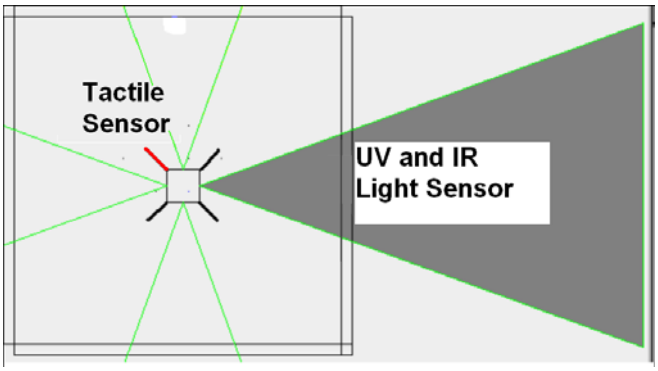


Fig. 3. The sensor base (small square in the center of the larger colony space square) can have tactile sensors attached at each corner with their presence and orientation learned by the genetic algorithm (GA). The range of the tactile sensors is fixed. UV and IR light sensors can be attached at the midpoints along the sides of the sensor base, with their presence, orientation, and range learned by the GA. The spread of the light sensors is fixed.

The IR sensor and the UV sensor range and spread overlap in the diagram. The light sensors are placed at the sides of the sensor base at any angle relative to the robot's heading. The tactile sensors are placed in the corners of the sensor base at any angle relative to the robot's heading. The sensors are all binary; they either detect a stimulus or do not. The learning system determined which sensors would be activated and their range (for light) and orientation. The spread of the sensors was not evolved since no mechanism was in place to adjust this aspect of the sensor. 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 actual robot.

The controller of the ServoBot is a reactive system that uses 13 rules of the form: if (sensor A detects a stimulus) then (trigger gait number X). Each sensor (with a maximum of 12) is associated with a specific rule and a single gait. There is also a rule 13, which is fired when no sensors are triggered, providing a default gait.

III. Evolution of Morphology and Control

The genetic algorithm (GA) was to learn the sensors to be used, the placement of these sensors, the range of the light sensors, and the consequents of the control rules described in Section 2.

A. Simulation

A 300 x 300 unit simulation area was used for learning. Fig 4 shows this area for the Central Mountain configuration. Please note that this diagram is rotated 90 degrees counterclockwise in relationship to the photograph in Fig 1. The top left corner of the simulation area was marked as the coordinate position (0, 0).

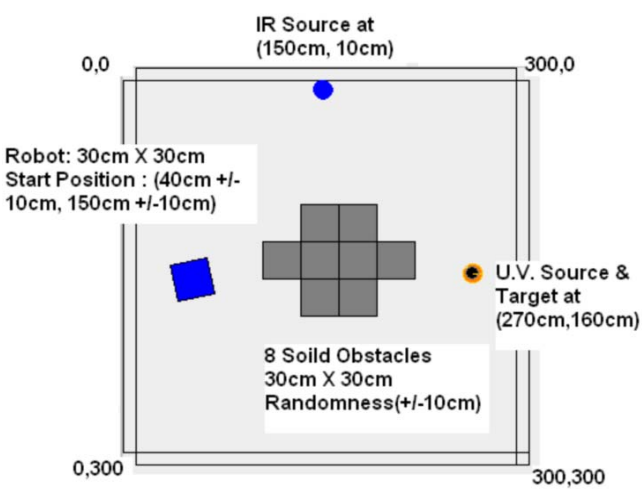


Fig. 4. Simulated environment used for learning (Central Mountain Configuration).

The simulated robot closely models the ServoBot with 16 possible gaits. The resultant movement of the robot after a single step of each of these gaits was measured and stored. Given a start position and orientation, each gait corresponds to different end positions and orientations after a single step. As the locomotion of the ServoBot is non-deterministic due to inconsistencies in its build and environmental variances, the turn values were randomized by +/- 0.2 units in the simulation. Sensor operation was simulated by assuming that any stimuli in the area of coverage of the sensor was sensed. The area of coverage for the tactile sensors was a straight line, 25 units in length. The area of coverage of the light sensors was a set angular span with the distance determined by the GA.

B. Chromosome and Genetic Operators

The GA population consisted of 256 individuals (chromosomes) that were randomly generated at the start of learning. A 212 bits long chromosome, divided into 35 sets (1 set of 12 bits, 12 sets of 9 bits, 12 sets of 5 bits and 13 sets of 4 bits), was used. The first 12 bits (1 set of 12 bits) represented which of the 12 sensors to keep running during the length of the run. The next 108 bits (12 sets of 9 bits) represent the angles at which each of the 12 sensors are to be placed onboard the robot base as all the sensors can be rotated 360 degrees (512 values per sensor, if values higher than 360 are chosen the program uses the chosen number minus 360). 40 bits (12 sets of 5 bits) represent the range of all of the light sensors (32 values for each light sensor). The last 52 bits (13 sets of 4 bits) represent gaits which are consequents for each of the 13 rules. The position of the sensors on the base are fixed as shown in Figures 2 & 3 and do not change.

The GA was run for 512 generations for each of the environment configurations. During evolution, all of the 35 sets of bits of the chromosomes underwent a single point crossover using stochastic (roulette wheel) selection of the parents. Rule selection had a 1% mutation rate. The other parameters had a mutation rate of 1% if any individual in the generation reached the goal and 6% if none of the individuals reached the goal.

C. Fitness

An agent successful in finding the target was assigned a fitness based on the number of sensors it had off and the amount of time it took the agent to get to the target, plus a bonus. To achieve the maximum fitness, the robot has to have all its sensors off and reach the goal without any time being used, giving it a theoretical maximum fitness of 15600 (this scenario is impossible). The fitness of an unsuccessful agent is dependent on how far away the robot is from the goal. The farthest possible distance in the robot colony from the goal is 313.8 cm. Twenty times this is 6276. An unsuccessful agent was awarded a fitness of 6276 minus 20 times its distance to the goal when the test was completed (Fig 5).

```
If (Agent_Reached_Goal)
  Fitness =  50 * Number_Sensors_Turned_Off  +
             50 * (Total_Time –
                   Time_to_Achieve_Goal) +
             Goal_Bonus
Else
  Fitness = 6276 –
           (20 * Resultant_Distance_from_Goal)
```

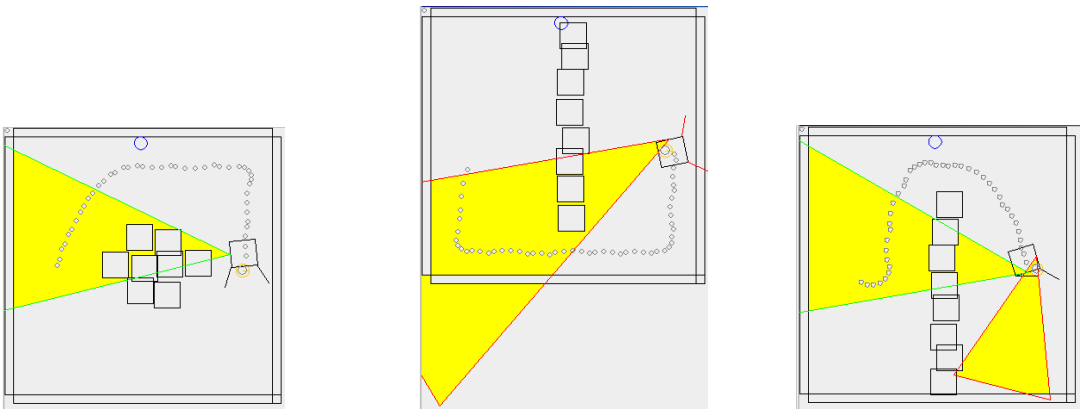
Fig. 5. The fitness function

Each individual had 3 chances, starting at a random heading and positioned within +/- 10 of the start position of

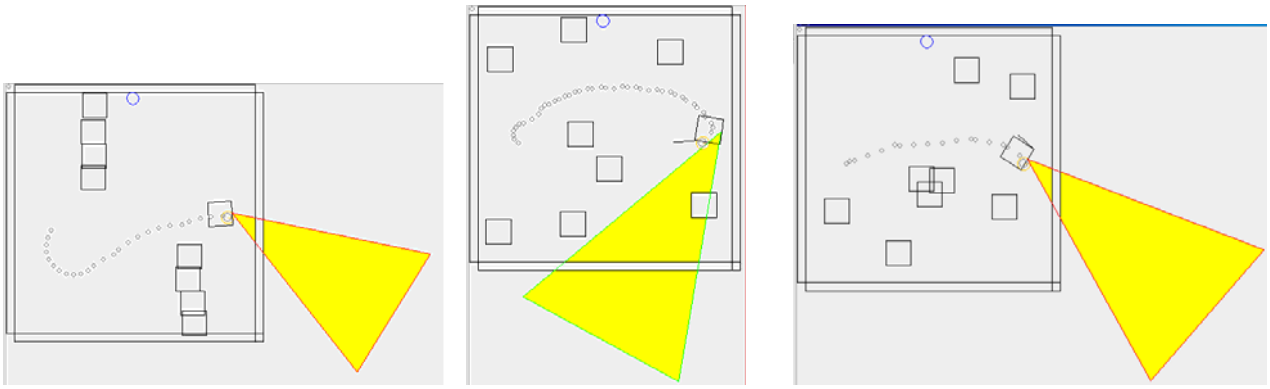
(40,150). The simulated robot agent had 300 steps to complete the task. A step for the ServoBot is defined to start with the legs in a ready to step position (right front, right back, and left middle legs forward, and the remaining legs back) and returns to this position after a full step cycle. Each of the 16 gaits completes a cycle in the same time. The time intervals required for each step continue even if a collision occurs. Since the robot has continual motion and a non-deterministic gait, it can work its way out of a collision. The run is stopped if the target position is reached or after 300 steps if it is not. After the entire population of individuals was tested and assigned a fitness, the individual with the highest fitness was added to the next generation without change. All of the individuals in the population were used for stochastic selection, with the most fit having the best chance of parenting an individual for the next generation.

IV. Simulation Results

Learning was done in simulation on seven environment configurations; those described in Section II, plus an additional one that was randomly generated. The entire learning process was repeated 5 times with random starting populations to check for consistency of the results. The learning trends (average and best individual of the population, over 5 runs, for each generation) for six of the environment configurations showed that the GA evolved solutions that quickly increased in fitness and continued to produce increasingly better solutions over the 512 generations. The seventh configuration (Random Two) was found to be too difficult for the learning system and the GA could not converge on a solution. Since a solution for this configuration was not found by the learning system, it was not used for the tests done on the actual robot.



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



Double Ridge:	Random One:	Random Three
1 Light Sensor Used at 22l; Length 140cm 2 Gaits and Rules Used	1 Tactile Sensor at 78 1 UV sensor at 21 3 Gaits and Rules Used	1 Tactile Sensor at 187 1 UV sensor at 9; Length 308 cm 3 Gaits and Rules Used

Fig. 6. Simulated results of the environments with successful solutions found, showing the selected sensors and the paths taken by sample solutions after 512 generations of learning.

Observation of the six successful final solutions in simulation showed that reasonable sensor configurations and controllers were produced that were appropriate for the type of environment. Figure 6 shows a sample from each of these six environment types. None of the solutions made use of the IR light. Since the UV light could be sensed over the obstacles, the learning system used UV sensors positioned at angles off from the robot heading to help position to robot to avoid the obstacles. This was not what we expected. We thought the system would use the IR light to maneuver into a position where it could turn directly toward the UV light. The learning system developed a more efficient method than we envisioned. The strategies used were a combination of finding orientation by using the UV light, wall-following, and tracking directly toward the target (UV light).

V. Tests on Robot

The strategy of evolving the sensor morphology with the control showed its success in producing a robot system that could navigate though most environments in simulation. However, the simulation is an ideal world where there the noise is randomly generated and the stimuli and the sensors are ideal. The results of the learning system needed to be tested in the real world (colony space in Fig. 1) to show the system's success. The main differences in the simulated environment and the real environment are that the light source is not ideal and the light sensors are not perfectly calibrated, the surface of the colony is carpeted making results of the gait steps of the ServoBot uncertain, and the tactile sensors are prone to noise due to the walking motion of the robot.

A. Tests

Each evolved sensor morphology along with its controller was transferred to the actual hexapod robot. The ServoBot was configured with the sensors placed as designated by the learning system and the learned control program was downloaded into the BASIC Stamp controller. The robot was then placed in the robot colony to test its ability to complete the navigation task. For these tests, the ServoBot was given a maximum of 200 steps to complete the task. As in the simulation, the obstacle placement in the environment had a degree of randomness of +/- 10cm along the x and y coordinates and the robot's start location was also +/- 10cm and the heading random. Also as in the simulation, a collision did not stop a test run since the ServoBot can work its way out of a collision due to its non-deterministic gait.

B. Results and Discussion

The sensor morphology and controller evolved for the Central Mountain, Double Ridge, Single Right Ridge, and two randomly generated environments were successful in completing the navigation task. When the paths are compared to the paths taken in simulation the results are very similar as can be seen in the time lapse photos of the Central Mountain and its simulated counterpart in Figure 7.

Although the solution found in simulation was successful on the actual robot for these five environments, the solution evolved for the Single Left Ridge configuration was not robust enough to transfer to the real world. The robot's failure to complete the task in the Single Left Ridge Environment showed that the success was also dependent on the simplicity of the design. In the case of Single Left Ridge, the final design evolved required 4 sensors and 5 rules. This was the most number of sensors required by any agent in all of the environment configurations. The dependence of the reactive controller on the reliability of the sensors means that the actions of the agent will be increasingly non-deterministic as the sensor input increases. Moreover, as the number of sensor inputs increase, the number of noisy actions increase, making it difficult for the robot to navigate through the given environment. The other designs, although subject to the similar noise in the environment were simpler in terms of controller complexity (i.e. number of rules used) and sensors used. The morphology / controllers learned were configured for their specific environment. In additional tests, the robot configured for one environment type was tested in another. These tests showed that the robot controllers specialized for one environment could not navigate other environments.

The test results show that the co-evolution of sensor morphology and controller for the ServoBot is a viable option for designing robust system to perform tasks in specific environment configurations. Unfortunately, our current system does not work in all situations. However, we believe that adjustments in the noise level or possibly a uniform degradation of the performance of all the simulated sensors would rectify this issue. Even though the sensor morphologies and controllers were evolved in a specific environment with a comparatively very low noise level, in most cases the designs produced were robust enough to perform well in the highly noisy real world environment. This is particularly of note since mechanical noise and environmental noise were not factored into the GA during the evolution process. Another important factor that the learning system addressed was the efficiency in terms of required sensors. There is a fine line between using a sensor that is only required some of the time and deleting a sensor to increase the efficiency. A major factor in the performance of the robot is the randomness introduced into the simulation during the GA learning. Controllers learning in an environment of uncertainty will probably be more robust in the noisy real world environment. Nevertheless, too much noise will prevent convergence and may force the system to add unneeded sensors.

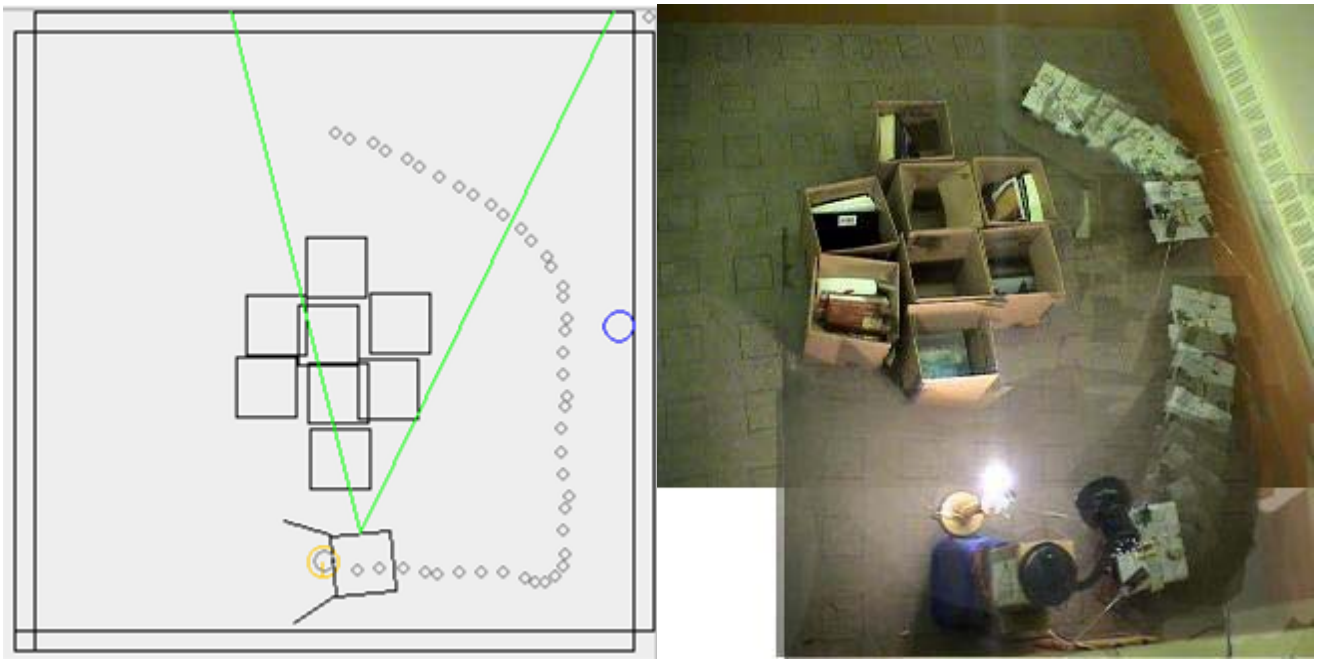


Fig 7 Comparison of simulated path to the actual performance.

VI. Conclusions

This paper presents research where a system of concurrently evolving the sensor morphology and control for a hexapod robot could be successfully done in simulation with the results transferrable to the actual robot. The results from the simulation and tests on the robot show that this approach provides an automated design process with designs that take into consideration dependencies between many variables, exploit the environment to complete the assigned task, determine what information is relevant and discard what's irrelevant, and can be successfully transferred to the real world despite noisy sensor data. The genetic algorithm, due to its design and dependency on the fitness function, inherently takes into consideration many variables whose dependencies do not have to be explicitly defined. Since these dependencies are inherent to evaluation of the fitness function, they are implicitly taken into consideration and adjusted. The evolved sensor morphologies and controllers were specialized for types of environments, but due to the randomized placement of the obstacles and randomized start position and orientation of the robot, the result produced is a generalized solution effective in the given environment. The solutions evolved were highly specialized in that they were not successful in the other environments.

There are many possibilities for future work. Experiments with varying degrees of noise added to the simulation environment or a reduction in the simulated sensor capabilities could help to ensure that all simulated results can be successfully transferred to the actual robot. Increasing the potential for a more complex controller, one that is more than reactive control, could help ensure that the learning system finds a solution in all environments where one is possible. The simple if…then rules that were used were useful for us to measure the complexity of the controllers, but limit the complexity of control that the system can attain. Tests could also be done on a greater variety of environments with differing tasks.

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] K. Suguira, "Exploiting Interaction between Sensory Morphology and Learning" 2006

[3] L. Linchentseiger, "Towards optimal sensor morphology for specific tasks: Evolution of an artificial compound eye for estimating time to contact" 2000

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

[5] A. Mark, D. Polani, and T. Uthmann. "A Framework for Sensor Evolution in a Population of Braitenberg Vehicle-like Agents," Artificial Life VI, pages 428--432, 1998

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

[7] 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.

[8] K. Suigua, T. Shiose, H. Kawakami, and O. Katai. "Co-evolution of Sensors and Controllers," Transactions of the Institute of Electrical Engineers of Japan. Vol.124; No.10; 2004.

[9] G.Parker, P. Nathan "Co-Evolution of Sensor Morphology and Control on a Simulated Legged Robot," Proceedings of the 7th IEEE International Symposium on Computational Intelligence in Robotics and Automation, 2007.

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