EVOLVING ROBOT BRAINS AND BODIES TOGETHER: AN EXPERIMENTAL INVESTIGATION

Wei-Po Lee

ABSTRACT

In recent years, evolutionary algorithms have been proposed to synthesize robot controllers. However, admittedly, it is not satisfactory just to evolve the control system, because the performance of the control system depends on other hardware parameters, the robot body plan. In this paper, an evolutionary framework is presented to evolve complete robot systems, including controllers and bodies, to achieve fitness-specified tasks. In order to assess the performance of the developed system, we use it to evolve controller and body plan together for a robot in a simulated environment to achieve an obstacle avoidance task. Experimental results show the promise of this hybrid evolutionary approach. In addition, the importance of simultaneously evolving controller and morphology is emphasized, and the distribution of body parameters in the morphological space is also studied.

Key Words: evolutionary computation, autonomous robots, artificial life, evolutionary robotics.

I. INTRODUCTION

Artificial life research complements traditional biological science that concerns analyses of living systems, as it synthesizes artificial systems that can exhibit life-like behaviors. In recent years, artificial life techniques (i.e., evolutionary algorithms) have been proposed to synthesize control systems for robots (Harvey *et al*., 1997; Lund and Asada, 1999; Nolfi and Floreano, 2000). The central idea is that by using the characteristic of self-adaptation in evolutionary algorithms, we can synthesize robot behaviors closer and closer to what we expect, over the generations, through fitness improvement.

Typically, in the work of evolving robot controllers, the robot is always assumed to have a non-reconfigurable physical structure and the evolutionary algorithm is applied to evolve control systems on this fixed robot structure for target tasks (Floreano and Mondada, 1996; Floreano and Urzelai, 2000; Harvey *et al*., 1997; Miglino *et al*., 1996; Nolfi, 1998). The process of evolving controllers is similar to traditional evolution-based work. A fitness function is first specified to quantitatively describe the desired robot behavior, and then the evolution process is begun to find the appropriate controllers.

In addition to the controller, as we can observe, the robot morphology, or body plan, also plays an important role in generating robot behaviors. The body plan here means the physical structure of a robot; it can be specified as a set of structure parameters. In the case of a mobile robot, for example, they may be the wheel radius, motor time constant, the number and position of sensors, and so on. In fact, the biological data in nature suggests that body plans of all animals are controlled by *Hox* genes (Day, 1995). It has been proposed and shown that an axial *Hox* code determines the morphology of individual vertebrates (Kessel and Gruss, 1990), and that this code plays a critical role in head development in both mice and *Drosophila* (Frasch *et al*., 1995). These evidences indicate that the control system and the morphology evolve together over the generations, and that the evolution of body plans can be modeled by having parts of a genetic string to express growth of different body parts.

Yet, the evolution of body plans has previously been ignored. Though Sims has used Lindenmayer's *L*-system (Lindenmayer, 1968) to evolve both the controller and 3D-rigid parts of a creature acting in a

W. P. Lee is with the Department of Management Information System, National Pingtung University of Science and Technology, Pingtung, Taiwan 912, R.O.C. (Email: wplee@mail.npust.edu.tw)

virtual poly-world (Sims, 1994), impossible regions in the morphological space (Riedl, 1978) are not considered in Sims' work, so his approach tends to be unworkable in real robots. Funes and Pollocks used Genetic Algorithms to evolve buildable objects such as Lego bridges (Funes and Pollocks, 1999), but not complete robot creatures that include controllers and physical structures.

In this paper, we present a framework in which the controller and morphology of a robot evolve together. Consequently, such a robot creature can improve its performance through the change of its control system or body plan. In this system, the robot consists of two parts: a tree-like controller and a string of structure parameters. A Genetic Programming(GP) subsystem is implemented to evolve the controller and a Genetic Algorithm(GA) subsystem, to determine the values of parameters. To assess the performance of the proposed approach, we use the evolutionary system to evolve complete robot creatures to achieve an obstacle avoidance task to explore the simultaneous evolution of robot controller and morphology. We also conduct a series of experiments to verify that both controllers and body plans of the evolved robots have adapted to the specific task. In addition, the distribution of the evolved robot body plans in the morphological space is investigated and analyzed. The results show the promise of our system.

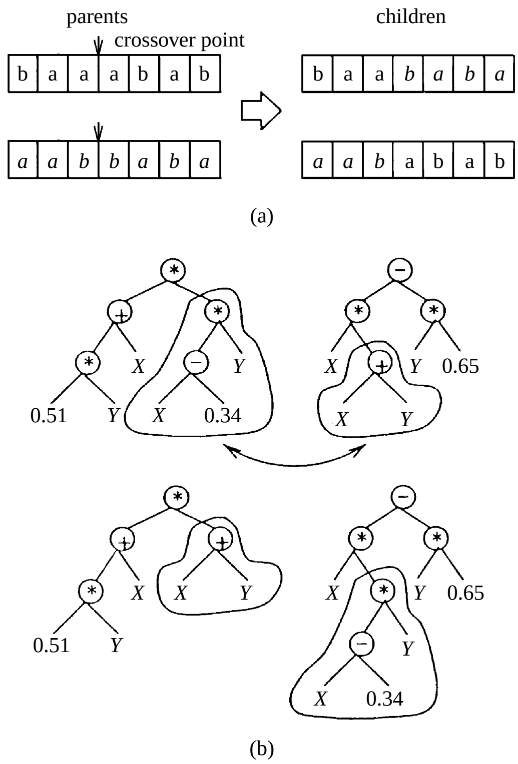

Fig. 1 (a) An example of one-point crossover in GA; (b) an example of crossover in GP

II. BACKGROUND

Genetic Algorithms (Mitchell, 1996) are currently the most popular forms in modeling the natural evolution process to solve problems with high complexities. In GAs, an individual is constituted by a string of genes; each of them is regarded as carrying a genetic feature. An individual with better fitness is said to have some genetic features that are capable of solving the problem, and these features are expected to be propagated to the subsequent population by means of duplicating or recombining the gene sequence of parent populations.

In general, an individual is represented as a fixed-length string, often, but not necessarily, in the form of a bit string. Most importantly, the design of a representation must ensure that a potential solution in the search space for the problem to be solved can be expressed as a string of the developed representation. This requires considerable knowledge of and insight into the problem.

GA is operated as an iteratively cyclic mechanism which includes a sequence of selecting parents and creating children, after the initialization phase. Selection involves probabilistically choosing individuals from the current population as parents to generate offspring to form a new population; and recreation is to apply genetic operators, such as reproduction, crossover, and mutation on the selected individuals to generate new ones. Among different operators, the crossover is the major one to create most of the offspring, and it is generally implemented as alternately copying gene sequences, separated by randomly chosen crossover points from the selected parents. Fig. 1(a) shows a typical example of crossover in GA.

A variant of Genetic Algorithms, named Genetic Programming, was invented by John Koza (Koza, 1992), and its popularity is now increasing in the community of evolutionary computation research. It is an extension of the traditional GAs with the basic distinction that the individuals are dynamic tree structures rather than fixed-length vectors. GP aims to evolve dynamic and executable structures often interpreted as computer programs to solve problems without explicit programming.

As in computer programming, a GP tree structure is constituted by a set of non-terminals as the internal nodes of the tree, and by a set of terminals as the external nodes (leaves) of the tree. The construction of a tree is based on syntactical rules which extend a root node to appropriate symbols (non-terminal and/or terminals) and each non-terminal is extended again by suitable rules accordingly, until all the branches of the tree end up with terminals. Hence, the first step in applying GP to solve a problem is to define appropriate non-terminals, terminals and the syntactical rules associated for the program development. The search space in GP is the space of all possible tree structures composed of non-terminals and terminals.

As in GAs, operators of reproduction, crossover, and mutation are used to create new tree individuals. Reproduction simply copies the original parent tree to the next generation; the crossover randomly swaps subtrees from two parents to generate two new trees; and mutation randomly regenerates a subtree from the parent to create new individuals. Because of such specific structures in GP, when crossover is performed, all syntactic constraints must be satisfied to guarantee the correctness of new trees. An example of GP crossover is illustrated in Fig. 1(b).

The process of evolving controllers is similar to traditional evolution-based work. It first specifies a fitness function which quantitatively describes the desired robot behavior, and then begins the evolution process: generating an initial population consisting of different control systems; evaluating the control system on a robot (simulated or real) to determine the corresponding performance; and applying genetic operators on the current robot population to create a new population, according to the fitness. By the self-adaptation of the internal control structure, the robot is expected to generate behaviors closer and closer to what is desired. The technique of evolutionary computation has been employed to develop robot controllers: in (Floreano and Mondada, 1996; Harvey *et al*., 1997), the authors used a GA-based system to evolve neural networks as controllers; in (Colombetti *et al*., 1996) GA was used to evolve classifier systems; and in (Nordin and Banzhaf, 1997) GP was used to evolve tree-like programs for robot control.

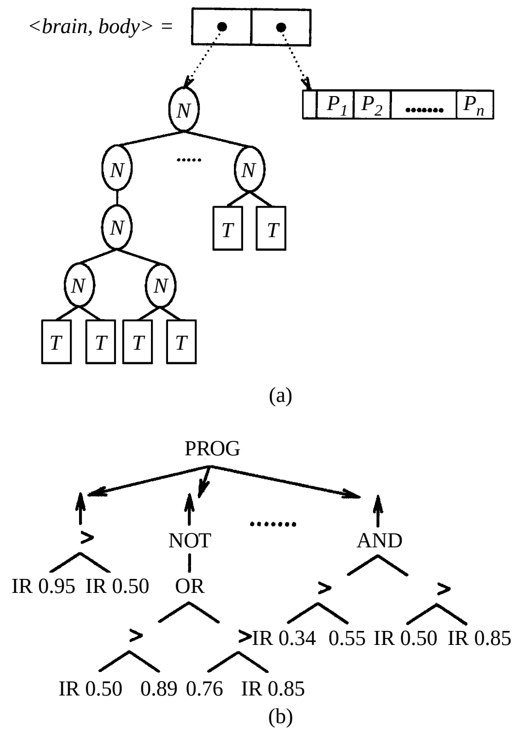

Fig. 2 (a) The structure of an individual defined in this work: in the tree structure, a node with an *N*/*T* is a non-terminal/terminal node; in the string representation, P_i is a real number; (b) Diagram of a typical controller in which the outputs of the subtrees (with arrows upward) are interpreted as motor commands to drive actuators. The node PROG is a dummy root node of a circuit tree

III. EVOLVING A COMPLETE ROBOT SYSTEM

In our evolutionary system, an individual is a complete robot creature; it consists of a control system and a body plan, represented as *<brain, body>*, in which the brain is a tree-like controller and the body is a string of real numbers quantitatively specifying the physical structure of a robot. The main reason we use a hybrid GP/GA approach is that we prefer a tree controller since it operates a variable-size genotype. This is an important feature for evolving control systems because it provides complete freedom for the control architecture in respect to the task complexity that is generally difficult to predict. On the other hand, it is quite straightforward to regard the structural design of a robot as a problem of parameter optimization and then employ GA to find the most appropriate combination of structural parameters. With this direct encoding, the model of the robot structure becomes relatively simple and the computational cost can be greatly reduced. Therefore, we choose to use a hybrid GP/GA representation for a robot individual. The typical structure of an individual is shown in Fig. 2(a).

The main flow of our hybrid system is similar to those of evolutionary robotics systems without an evolving body plan. After an environment has been given and a goal formulated as a fitness function, an initial population is created at random. But here, each individual has its own brain and body. To evaluate an individual is to test the brain with the corresponding robot body for a period of time, in the given environment, and to measure the performance. An individual's survival probability is determined by how well the controller performs with the corresponding body, to fit the evaluation criterion. Like a conventional evolutionary system, after evaluating each individual, a certain selection method is employed to choose parent individuals and then genetic operations are applied on them to create offspring. Genetic operations, such as reproduction, crossover, and mutation, are used in this work. The reproduction operation simply copies the selected parent individuals, without changing the controllers or the bodies, into the next generation. The crossover or mutation operation is allowed to take place on the brain(s) or the body (bodies) at random. Because of the special structure of an individual here, whenever a crossover or mutation is used to create new individuals, a random selection of brain-body is done first to determine which part (brain or body) the operator should apply to. Then the corresponding operation is performed on the part(s) of the parent(s) chosen. In this way, the crossover can be constrained to occur on both brains or both bodies of the involved parents in order to maintain the correctness of the structures. Related techniques of GP and GA are applied independently for the brains and the bodies.

A brain here means a reactive or sequential behavior controller, depending on whether internal states are involved. In this paper, we focus on evolving reactive behavior controllers; the same approach can be extended to evolve sequential ones. We regard a reactive behavior controller as a combinational circuit network. By duplicating and separating those components, the output of which serve as inputs of multiple components, and by introducing a dummy root node to connect the outputs of a circuit network together, it is very straightforward to convert a circuit network to a tree. We then use such a tree to represent a controller in our work.

In our representation, we structure the perception information into *sensor conditionals* and use them as the inputs of a logic circuit. In our controllers, three types of non-terminals are defined: the dummy root node, the comparator, and the logic components. The dummy root node is to collect the main outputs of a control system for convenient manipulation by a GP system; the comparator is to construct the sensor conditionals; and the logic components are to constitute the main frame of the controller for mapping the structured sensor information into appropriate actuator commands. Terminals include two types: sensor terminals and numerical thresholds which constitute the sensor conditionals. A sensor conditional has a constrained syntactic structure; it exists in the form of $X>Y$, where X, Y can be any normalized sensor response or numerical threshold which is determined genetically. Fig. 2(b) illustrates an example of a robot's brain. In such a structure, the outputs of the sub-trees of a circuit tree are interpreted as commands (i.e., mapped into a table of motor commands) to drive actuators.

In our work, a robot is assumed to have a round physical body with sensors positioned around the body pointing radially outward. Depending on the characteristics of the specific tasks, different kinds of sensors are required. A sensor is defined to be associated with a value between 0 and 1 (this value is randomly generated in the first generation and then can be changed by crossover and mutation during evolution) which indicates the angle between the direction the sensor is pointing at and the heading of the robot. Whenever a sensor is called in the control system, the normalized sensor response is returned, in the direction indicated by the value associated with that sensor. For instance, an infra-red sensor with a value 0.3 (written as "IR 0.3" in a sensor conditional) will return the normalized sensor response in the direction 0.3 revolution (108 degrees) anti-clockwise, relative to the heading of the robot. In this way, the sensor positions/directions are also allowed to evolve with the controllers. More details about the development of our controller can be found in (Lee, 1999).

In order to evolve a robot body plan we need to analyze what constitutes a body plan and to extract the determining elements, which affect the behavior of a robot profoundly, from the physical structure of a robot. In the mobile robot design, for instance, there are some determining elements such as the wheel radius, the width of the wheel base, the time constant of the motion system, the body size (the diameter of the body, if we assume the robot body is round), and positions (with orientations) of the sensors, etc. The wheel radius affects the speed of the robot and determines the maximum and minimum moving speed for the specified motor commands; the width of the wheel base determines the turning rate of a robot; the time constant affects the response of the robot and determines the acceleration of the robot; the size of a robot body should be task-oriented: to avoid obstacles it may need to be smaller but to push boxes it may need to be larger; and the positions and orientations of the sensors allow the robot to acquire the perceptual information it needs. To evolve a robot body plan, in fact, means to decide these determining structural parameters of a robot genetically.

In our system, the structural parameters (except, as we have seen, the sensor placement) are arranged as a linear string, in which each position is filled up with a real number representing the value of the corresponding parameter. Due to hardware limitations and performance considerations, each structural parameter has its own lower and upper bounds; when we build a robot, the value of each structural parameter must lie in its bounds. Thus, a robot body can be expressed as $P_1P_2.....P_n$ where $Min(P_i) \leq P_i \leq Max(P_i)$, and $1 \leq i \leq n$.

For this linear string, a Genetic Algorithm is employed to determine the value of each structural parameter P_i within its own range. The crossover here

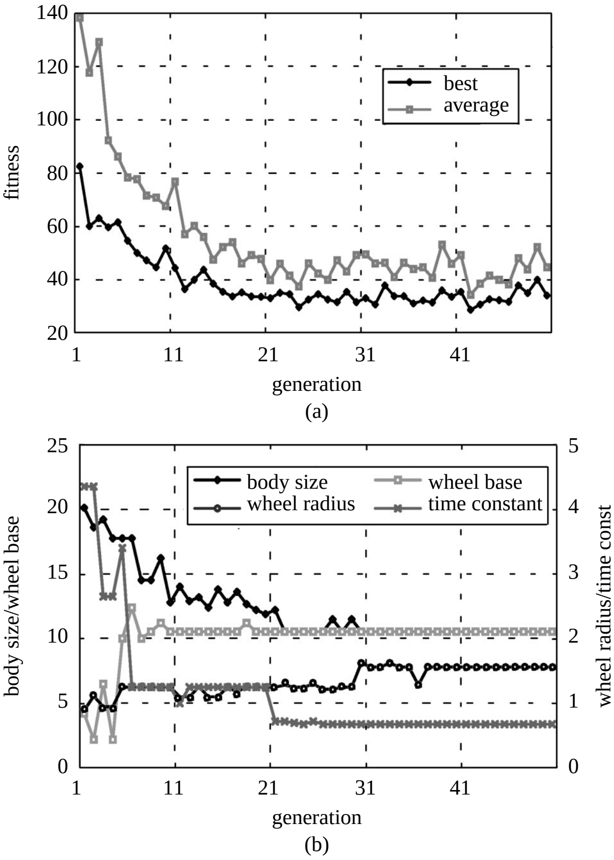

Fig. 3 (a) An example of the behavior of the evolutionary system, when it was used to evolve both robot brains and bodies to achieve an obstacle avoidance task; (b) curves show how the body parameters changed during the example run

is slightly different from the standard one: it is defined to perform operations of swapping or averaging randomly, for each pair of P_i between the two chosen crossover points. The reason for using the operation of averaging is to change the structural parameters gradually for fine-tuning.

IV. EXPERIMENTAL RESULTS

1. Evolving Brains and Bodies Together

In the following experiments, we use the evolutionary system presented above to evolve robot controllers and body plans together to achieve an obstacle avoidance task. The structural parameters we intended to evolve in this work were motor time constant, wheel base, wheel radius, and body size. Here, the value of the motor time constant was restricted to lie between 0.5 and 2.5 seconds; the lower bound and upper bound of the wheel radius were 1.0 cm and 3.0 cm; the value of body size was limited to be in the range 10 cm to 25 cm; and the wheel base was constrained to be no larger than the body size.

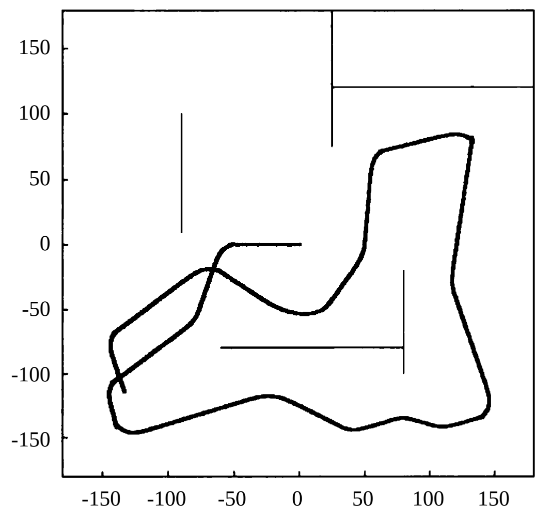

Fig. 4 Typical obstacle avoidance behavior

These values were chosen to approximate the device constraints of Lego robots, which provide the possibility of body reconstruction.

The obstacle avoidance task requires the robot to move forward as straight as possible but not bump any obstacle. Therefore, the fitness function was defined as the linear combination of the maximum of the IR responses, the normalized forward speed, and the absolute value of the normalized rotation speed (Lee, 1999). Fig. 3(a) illustrates an example of how our evolutionary system converged in evolving robot controllers with body plans. In order to examine how the structural parameters varied during the evolution, the corresponding curves are presented in Fig. 3(b). It shows the tendency of these parameters and indicates that the structural parameters reached some appropriate values quickly and then remained stable. In addition, we have also evaluated some of the best robot systems obtained from certain generations during the runs to observe which of the controller or body plan was evolved first. We found that the expected structural parameters seem to come out slightly earlier than the controller: the parameters were evolved to some stable values in generations 15 to 20, while the reliable and robust robot system was found in generations 25 to 30. This may be interpreted as the effect of the robot body plan being relatively straightforward, so, the inappropriate body plans are discarded soon. Though the structural parameters are not necessarily some definite values for a specific task, their values need to fit certain scalar relationships. The typical behavior of the evolved robot are shown in Fig. 4. As can be observed, a complete robot system has been evolved successfully.

We can also analyze the strategies the evolved robot uses to generate obstacle avoidance behaviors.

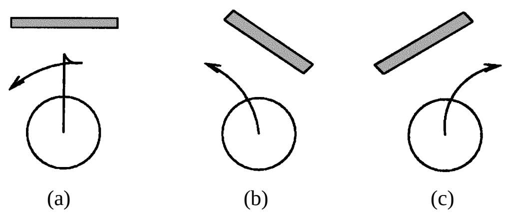

Fig. 5 Three different strategies used by the evolved robot to avoid obstacles

The sensor placements have been evolved with the controller to generate three ways to achieve the task: a front obstacle will cause the robot to move backward and turn to the right first, then move forward and turn to left to avoid an obstacle as indicated in Fig. 5(a); a left front obstacle will cause the robot to move forward with right turning as indicated in Fig. 5(b); and right front obstacle will cause the robot to move forward with left turning to avoid the obstacle as Fig. 5(c). The last two are similar to the strategy employed by Braitenberg's "fear" vehicle (Braitenberg, 1986). However, his vehicle will get stuck in front of a symmetric obstacle; which will not happen on our evolved vehicle because of the extra avoiding strategy shown in Fig. 5(a).

2. The Importance of Appropriate Brain-Body Coupling

We have shown that the controller and the robot body can be evolved together to achieve a fitness-specified task. In order to investigate how the evolved controller relies on the correspondingly evolved body (i.e., to prove they have been evolved together), we tested the evolved controller on different robot body plans, which were designed according to various combinations of structural parameters. Here, three different values for each structure parameter were used to generate different combinations for robot body plans.

We tested each body plan with the above evolved controller in 100 trials for different starting positions and changed environments. The results show that the combination where the controller was tested with the robot body it evolved together with has the highest success rate (actually it is 100%). The inappropriate brain-body couplings cannot achieve the task perfectly. This demonstrates that the brain and body of a robot have both participated in the evolutionary process and have adapted to the specific task.

3. Exploring Morphological Space

Some biological data and philosophical reflections suggest that in nature, different species cluster

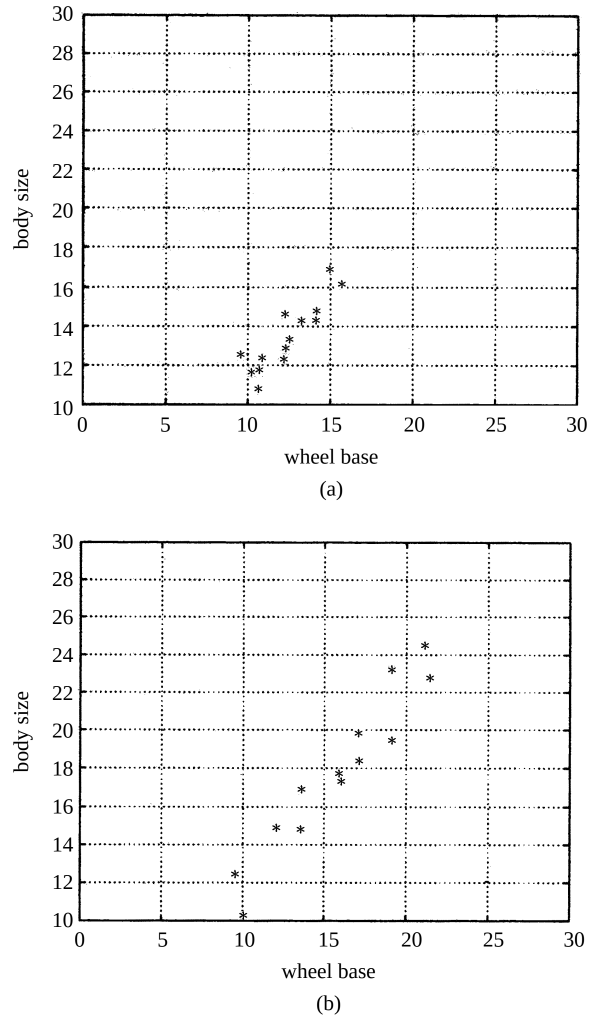

Fig. 6 The distribution of the body size and wheel base of the evolved robots. (a) the sensor range is 15 cm; (b) the sensor range is 30 cm

together in a morphological space and each individual cluster keeps a certain distance from others (Riedl, 1978). This implies that some forms are impossible in the morphological space; individual goals must have constrained the development of creatures' body plans to a certain extent. In this section, we conduct a series of experiments to study the distribution of the evolved body plans in such a morphological space. It not only allows us to compare the phenomena of species clustering in natural and artificial evolution, but also helps a robot designer to understand what features are especially desired and beneficial to the robot in achieving a specific task.

Figure 6(a) shows the distribution of body size and wheel base of the evolved robots in the morphological space, when they were evolved with controllers for an obstacle avoidance task. We observe that a robot with relatively small body size seems to be the fittest to perform the task and that there is an almost linear relationship between body size and wheel base. These results explain that a small size robot is relatively more capable of avoiding obstacles, and body-size wheel base can reduce the turning rate in order to keep the robot moving stably. Furthermore, when the sensor range was increased in another set of runs, as in Fig. 6(b), the correlation between the body size and wheel base remained linear, approximately, but the range of coverage of the evolved body size and wheel base became wider. This is because the robots could sense obstacles in the far distance, and therefore had more time to react. Hence, a larger body size and a larger wheel base that needs more time steps to turn the robot away from the obstacle can be evolved. In other words, the upper limit of the size and base of a robot is constrained by the sensor range.

In addition, Fig. 7(a) shows the distribution of the motor time constant and wheel radius of the evolved robot body plans. It can be observed that all of the evolved robots cluster in the region of extremely small motor time constant and large wheel radius. It indicates that in order to achieve the specific task successfully, the robot needs motors that have fast response so that it can sufficiently control its speed of moving and turning while frequently accelerating and decelerating. Also, relatively large wheel radii were evolved since they can give the robots inherent superiority in action.

Further, we used fixed large time constants, which implies slowing down the motor response, and then repeated the above experiment. Fig. 7(b) compares the results of three different criteria, with the rates of their correspondingly successful runs, 85%, 65%, and 55%. As can be seen, for those robots with fixed large motor time constants (slow motor response), small wheel radii are better choices for better control of the motion. Under such circumstances, the achievement of the specific task for the robot will rely heavily on the control systems. As a result, in comparison to the case of simultaneously evolving time constant and wheel radius, we find that the robots with fixed slow response cannot achieve the task easily: the rate of successful runs decreases obviously.

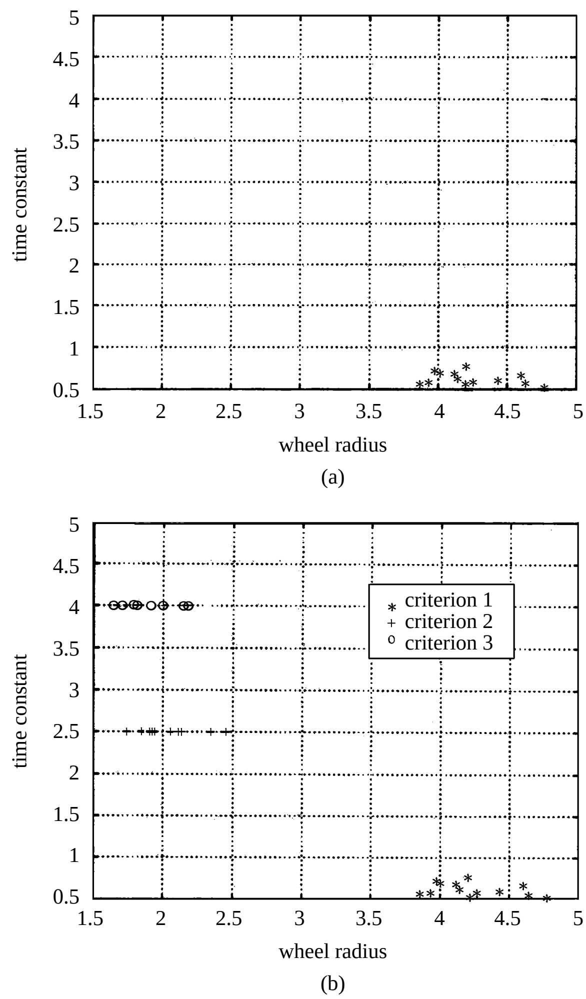

Fig. 7 (a) The distribution of the evolved motor time constant and wheel radius in the robot morphological space; (b) the distribution of the evolved body parameters in which criterion 1 evolved both the time constant and wheel radius with controllers; criterion 2 used fixed time constant (2.5 sec) and only wheel radius was evolved together with the controller; criterion 3 used fixed time constant (4.0 sec)

V. CONCLUSION

In this paper, we have presented a hybrid system of Genetic Programming and Genetic Algorithm to evolve complete autonomous robots. In our system, a circuit tree has been defined to represent a behavior controller for the robot, and the crucial structural parameters of a physical robot are extracted and arranged as a linear string of real numbers to represent a robot body. The GP and GA parts of the system can be used to determine the tree-structure of a controller and the string of structural parameters, respectively. To assess our evolutionary system, we use it to evolve a complete robot to achieve a fitness-specified task and evaluate the corresponding performance. The experimental results show the promise of our approach.

Also, we have analyzed the importance of appropriate brain-body coupling in designing a robot system. The evolved controller can achieve the task perfectly only when performing with the robot body plan it evolved together with. Finally, we have explored how the evolved robots cluster in the morphological space. Analysis of the robot morphological space indicates that some body plans are impossible or impracticable, so the evolved control systems are likely to fail if this is not taken into consideration. All the experimental investigations have shown that evolving both controllers and morphologies for robot systems provides a useful alternative for designing completely autonomous robot systems.

Our work presented here points to some prospects for future research. Because our experiments in evolving robot controllers with body plans are done in simulation, consequently more research into the realization of this approach in the real world can be undertaken. To achieve this, we can build a real Lego robot according to the evolved body specification, and download the correspondingly evolved controller to it to evaluate the performance. The information obtained from the observation of the performance difference between the simulated and real robots can be used to improve the simulator further, to bring our approach of evolving controllers and body plans together into the real world. Another possibility is to use the developed system to achieve a variety of tasks or more difficult tasks to examine its generality.

ACKNOWLEDGMENTS

This work was supported by the National Science Council of the Republic of China under Grant No. NSC89-2213-E-020-008.

REFERENCES

Braitenberg, V., 1986, *Vehicles: Experiments in Synthetic Psychology*, MIT Press, MA.

Colombetti, M., Dorigo, M., and Borghi, G., 1996, "Behavior Analysis and Training: A Methodology for Behavior Engineering," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 26, No. 6, pp. 365-380.

Day, S., 1995, "Invasion of the Shapechangers," *New Scientist*, No. 2001, pp. 30-35.

Floreano, D., and Mondada, F., 1996, "Evolution of Homing and Navigation in a Real Robot," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 26, No. 3, pp. 396-407.

Floreano, D., and Mondada, F., 1998, "Evolutionary Neurocontrollers for Autonomous Mobile Robots," *Neural Networks*, Vol. 11, pp. 1461-1478.

Floreano, D., and Urzelai, J., 2000, "Evolutionary Robots with On-line Self-Organization and Behavioral Fitness," *Neural Networks*, Vol. 13, pp. 431-443.

Frasch, M., Chen, X., and Lufkin, T., 1995, "Evolutionary-conserved Enhancers Direct Region-specific Expression of the Murine Hoxa-1 and Hoxa-2 Loci in Both Mice and Drosophila," *Development*, No. 121, pp. 957-974.

Funes, P., and Pollocks, J., 1999, "Computer Evolution of Buildable Objects," *Artificial Life*, Vol. 4, No. 4, pp. 337-357.

Harvey, I., Husbands, P., Cliff, D., Thompson, A., and Jakobi, N., 1997, "Evolutionary Robotics: the Sussex Approach," *Robotics and Autonomous Systems*, Vol. 20, No. 2-4, pp. 205-224.

Kessel, M., and Gruss, P., 1990, "Murine Developmental Control Gene," *Science*, No. 249, pp. 347-379.

Koza, J. R., 1992, *Genetic Programming: on the Programming of Computers by Means of Natural Selection*, MIT Press, MA.

Lee, W. P., 1999, "Evolving Complex Robot Behaviors," *Information Sciences: An International Journal*, Vol. 121, No. 1-2, pp. 1-15.

Lindenmayer, A., 1968, "Mathematical Models for Cellular Interactions in Development," *Journal of Theoretical Biology*, Vol. 18, pp. 280-315.

Lund, H. H., and Asada, M., 1999, "Embodied Artificial Life," *Artificial Life*, Vol. 4, No. 4, pp. 303-307.

Miglino, O., Lund, H. H., and Nolfi, S., 1996, "Evolving Mobile Robots in Simulated and Real Environments," *Artificial Life*, Vol. 2, No. 4, pp. 417-434.

Mitchell, M., 1996, *An Introduction to Genetic Algorithms*, MIT Press, MA.

Nolfi, S., 1998, "Evolutionary Robotics: Exploiting the Full Power of Self-Organization," *Connection Science*, Vol. 10, No. 3-4, pp. 167-183.

Nolfi, S., and Floreano, D., 2000, *Evolutionary Robotics: the Biology, Intelligence, and Techonology of Self-Organizing Machines*, MIT Press, MA.

Nordin, P., and Banzhaf, B., 1997, "An On-line Method to Evolve Behavior and to Control a Mininature Robot in Real Time with Genetic Programming," *Adaptive Behavior*, Vol. 5, No. 2, pp. 107-140.

Riedl, R., 1978, *Order in Living Organisms: A System Analysis of Evolution*, John-Wiley & Sons, NY.

Sims, K., 1994, "Evolving 3D Morphology & Behavior by Competition," *Proceedings of the Fourth International Conference on Artificial Life IV*, pp. 28-39.

Spirov, A. V., 1993, "Changes of Initial Symmetry in the Pattern-form Interaction Model of Sea Urchin Early Development," *Journal of Theoretical Biology*, Vol. 161, pp. 491-504.

Manuscript Received: Mar. 07, 2001
Revision Received: Feb. 01, 2002
and Accepted: May 27, 2002