AIBO: Toward the Era of Digital Creatures

Masahiro Fujita

The International Journal of Robotics Research 2001 20: 781
DOI: 10.1177/02783640122068092

Masahiro Fujita

Digital Creatures Laboratory
Sony Corporation
6-7-35 Kitashinagawa
Shinagawa-ku, Tokyo 141-0001, Japan
mfujita@pdp.crl.sony.co.jp

AIBO: Toward the Era of Digital Creatures

Abstract

The 21st century will become an era of autonomous robots that help and support people. Thus, they will be considered as partners of human beings. In this paper, the author introduces AIBO, the first product model of Robot Entertainment Systems. The main application of this robot is a pet-style robot, which must maintain a lifelike appearance. The author suggests how to maximize the complexity of responses and movements to solve the problem of substantially increasing the lifelike appearance of autonomous robots. The technologies used in AIBO are also described. Although AIBO is not intended for service or hazardous work, the development of AIBO is a major step toward a new era of autonomous robots in the new century.

KEY WORDS—autonomous robot, pet-type robot, behavior control architecture, complex behaviors, real-world agent

1. Introduction

We are advocating a new application field of autonomous robots, focusing on robots for entertainment purposes. Conventional autonomous robots have been proposed for use in service and dangerous work, but major technological hurdles must be overcome before robots are viable for mission-critical operations in these fields because mistakes in those application domains cannot be tolerated. However, when entertainment robots make mistakes, such as failing to correctly recognize objects, no life-threatening problems ensue. For this reason, this novel application area is fully viable even at the current technology level. We consequently decided to promote the field we call *robot entertainment* and have built a number of prototype robots (Fujita and Kageyama 1997; Fujita and Kitano 1998; Fujita, Kitano, and Kageyama 1998). These prototypes mainly used software designed for pet-style robots, and we studied what is important for this type of robot. We concluded that the critical requirements all converge on the problem of "maximizing the lifelike appearance" of the robot.

The difficulty with this problem statement is that there is not a good evaluation method for "lifelike appearance." Subjective evaluation with the semantic differential (SD) method (Shibata, Yoshida, and Yamato 1997) is one of the methods, however, in which evaluations must be done with many subjects with careful mental state control during the experience. It may be useful for final product evaluation, but during design and development periods, it is not a proper criterion because of the time-consuming evaluation process. Furthermore, the final design of behaviors and motions must be very relevant for a lifelike appearance. Therefore, we should concentrate not on the details of motions but rather on the mechanism of their generation.

We reformulated this problem as maximizing the complexity of responses and movements and worked from there. Of course, it is not an identical problem statement, but maximizing complexity is easier to evaluate than maximizing lifelike appearance. It is now also possible for us to discuss the mechanism of behavior and motion generation in this context.

An argument arises that the viewer's suspension of disbelief might be broken if the robot did something really stupid, such as walking repeatedly into a wall. From the viewpoint of complexity, however, the robot shows only a simple, single behavior, which is to walk into the wall nearly every time it finds itself in the same situation. If we can increase the number of behaviors exhibited in the same external situation, thus increasing the complexity, then a repeated behavior will not reappear. In addition, some technologies, such as introducing artificial instincts and emotions, with increasing the number of behaviors, further ensure realizing nonrepeated behavior exhibition. We will discuss this issue later.

In this paper, we do not provide a quantitative definition for "complexity of responses and movements" but rather suggest the introduction of the following factors as one way of assessing solutions to this problem. These factors are (1) multiple motivations for movements, (2) a configuration with high degrees of freedom, and (3) nonrepeated behavior exhibition.

Here, the word *motivation* has a similar meaning as for the motivation of animal behaviors (Haliday and Slater 1983). It is alternatively referred to as drives, emotions (Velasquez,

The International Journal of Robotics Research
Vol. 20, No. 10, October 2001, pp. 781-794,

Kitano, and Fujita 1998), or instincts. Naturally, the "instincts and emotions" model for this pet-style robot processes information in a style similar to the mammalian brain and takes account of biological behavior. However, our motive for introducing this model is not to see how well we can achieve mammalian instincts and emotions; rather, we want to use that emotional model to determine how to maximize the complexity of movements and behaviors of autonomous robots. We are not looking at the degree to which the robots resemble actual animals, but we are looking at the mechanism whereby we can use this model for maximizing the complexity of movements and behaviors.

Regarding the second factor, a configuration with a high number of degrees of freedom, the prototypes described in this paper resemble a small animal such as a dog. Of course, its shape must be important; however, we address here the importance of the robot possessing a high number of degrees of freedom. Even if it has four legs, if all the movements of the legs are achieved by only one degree of freedom each, the robot will not be seen as exhibiting a lifelike appearance.

These two factors are solutions to increasing the number of behaviors. However, maximizing the complexity of responses and movements does not mean only increasing the number of behaviors. Then, the third factor gives a solution to the problem of how to realize nonrepeated behavior exhibition. There are several ways to realize nonrepeated behavior exhibition. Introducing artificial instincts and emotions is our first trial. In addition, learning and development of behaviors through interactions with humans and environment are also effective.

The remainder of this paper first outlines the design concept for a series of prototype robots and the overall agent architecture of these pet robots. The first two factors mentioned above will be discussed in the agent architecture of the prototype robot. A part of the third factor, which is introducing artificial instincts and emotions, will be also discussed in the same section. Then we explain the technologies used in AIBO ERS-110, which include image processing, sound processing, and walking pattern generation. We again explain the emotions and instincts model used in AIBO. The remaining issue of the third factor mentioned above, which is learning and development, will be also discussed here. Finally, we report some facts of AIBO, including its marketing results, followed by comparison with other related works.

2. Pet-Style Robots

To reiterate, maximizing the lifelike appearance is considered the most important problem for pet-style robots. We have reformulated this problem as maximizing the complexity of responses and movements. This serves as our overall approach to configuring an autonomous entertainment robot. The main points involved are as follows:

1. A configuration of four legs, each of which has 3 degrees of freedom; a neck with 3 degrees of freedom; and a tail with 1 degree of freedom. Altogether, this amounts to 16 degrees of freedom. With such multiple degrees of freedom available for motion generation, the complexity of movements is increased.
2. The generation of multiple motivations, the generation of behaviors based on the motivations, and selection among the behaviors. There are a large variety of combinations of behaviors, and this exponentially increases the complexity of observed behavior. The behaviors are generated from the following:
   (a) a fusion of reflexive and deliberate behavior over a ranging time scale;
   (b) a fusion of independent motivations given to the robot parts, such as the head, tail, and legs;
   (c) a fusion of behaviors that obey both external stimuli and internal desires (instincts, emotions).
3. The internal status (instincts and emotions) changes the behavior of the robot toward external stimuli. Furthermore, the internal status can change according to external stimuli. Thus, the overall complexity of overt exhibited behavior is increased.
4. Adaptation through learning is introduced, so that the degree of complexity is increased when the robot is observed over a long period of time.

Figure 1 shows an example of a prototype four-legged robot, named MUTANT, while Figure 2 shows the mechanical configuration and sensors that the robot is equipped with.

This robot uses 16 servomotors, each composed of a DC geared motor and a potentiometer, to enable flexible movement. The robot is programmed to react to the external world

Fig. 1. The pet-style robot MUTANT.

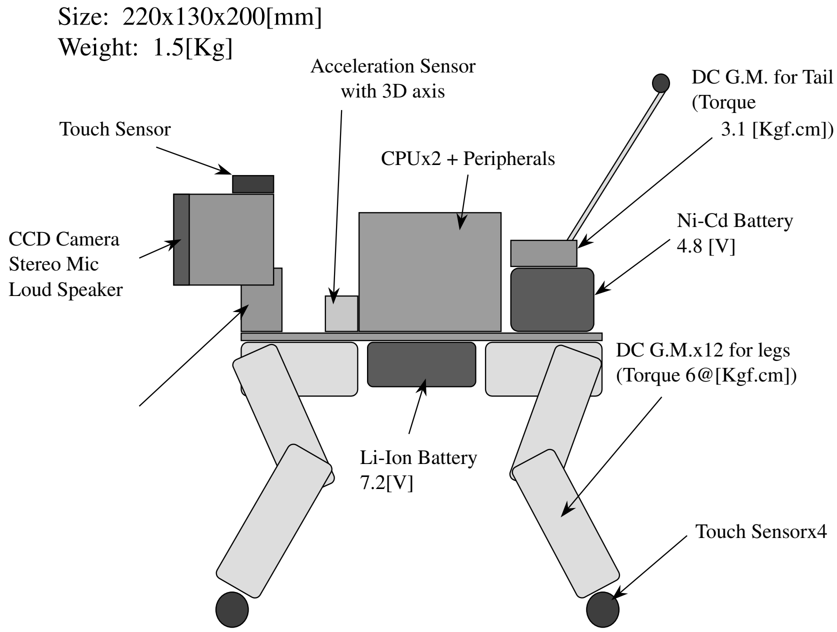

Fig. 2. Mechanical configuration and sensors of MUTANT.

and to humans by using its capacity for expression while employing a variety of sensory processing. The aim is to give the impression that the robot is alive. It is equipped with a micro-CCD camera, a stereo microphone, and an acceleration sensor (three-axis), and it can perform image processing, acoustic signal processing, and position estimation. For example, as shown in Figure 3, its basic movements include the following:

1. searching for a colored ball, then approaching it and kicking it;
2. expressing its simulated emotional state, such as being "angry";
3. giving its paw;
4. sleeping when it gets tired.

2.1. Agent Architecture for MUTANT

As described in the previous section, we incorporate some behavioral design principles within the architecture. The first aspect is the use of both reflexive and deliberate behaviors, which can be considered as distributed along a time-scale axis. The reflexive behaviors should be handled in a very short time, but the deliberative behaviors can be executed more slowly and carefully. An example of a reflexive behavior is the response generated when a user hits the head of the robot: the robot shakes its head while generating a sound designed for an astonished situation. In this robot's case, the visual tracking behavior is also a reflexive behavior. Usually, to track a ball smoothly, it is necessary for the robot to update the position of the ball in less than 60 msec. On the other hand, it takes a long time, when compared with this reflexive behavior, to process and interpret a sound command of tone signals that are formed by music chords in arpeggio. This is considered a deliberative behavior.

From an engineering point of view, the computation time, or minimum feedback latency for a response, is a practical measure for distinction between reflexive and deliberate behaviors. However, we believe that there is a more fundamental distinction, which is computational complexity of responses. Reflexive behaviors must decide actions corresponding to external stimuli as soon as possible; therefore, a decision rule or decision function must be a simple noniterative mapping. On the other hand, some decision rules must be searched for answers in a huge space of the database. In general, behaviors with planning or inference, which need to search for an answer in the rule database, can be considered as deliberate behaviors. In this paper, we will not use such deliberate behaviors. Instead, we use the computation time to distinguish reflexive and deliberate behaviors.

To handle the fusion of reflexive and deliberative behaviors, we employ the layered architecture shown in Figure 4. The lowest layer is the motor command generator, which produces motor commands using sensor feedback and top-down commands from the upper layer. Reflexive behaviors are generated at this level. For example, a touch sensor signal located on the head is used in this layer, which generates a head-shaking action. In another example, the position data of

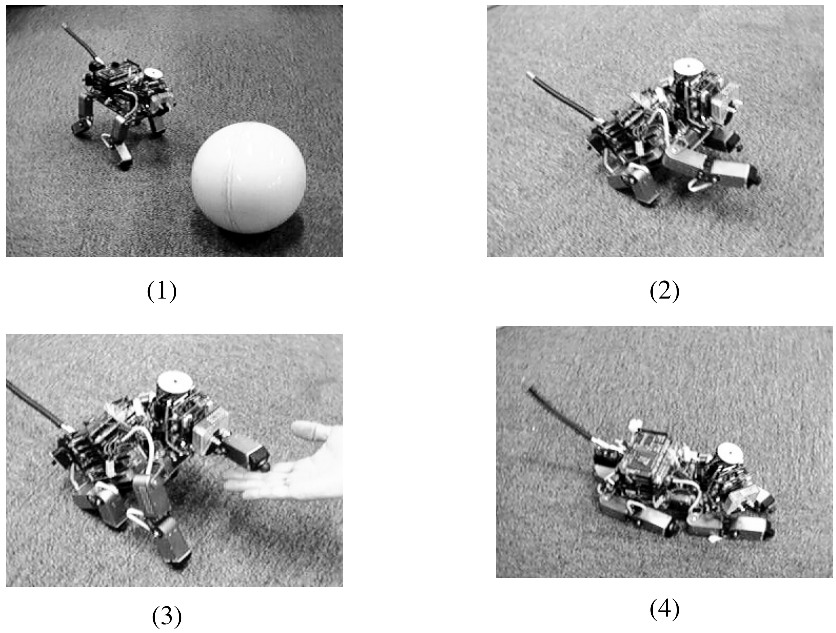

Fig. 3. Diverse movements (see text for description).

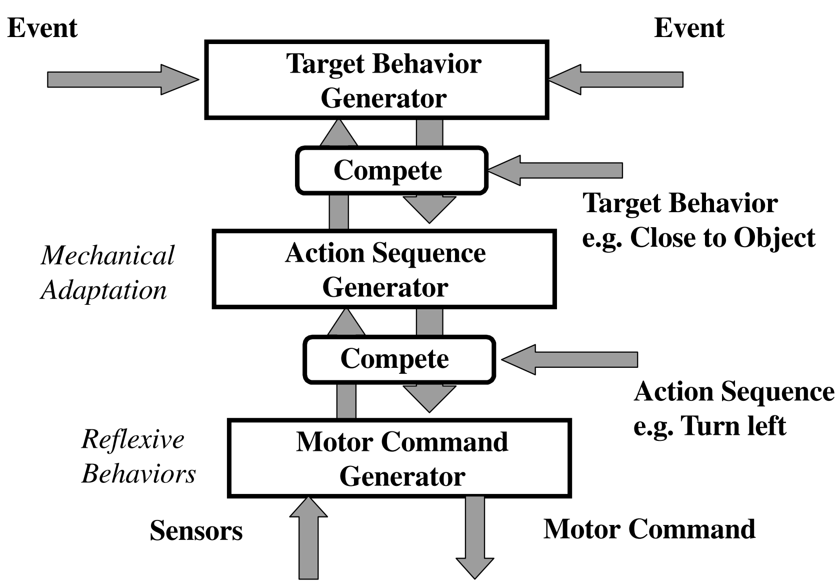

Fig. 4. A layered architecture.

the tracked ball are used at this level and generate the motor commands for tracking.

The middle layer is an action sequence generator that, as the name implies, generates action sequences forming a behavior. For example, assume that the quadruped robot is in a sitting posture, and the upper layer sends the command to approach the ball. Then, the robot has to first stand up, then start walking toward the ball. Thus, two actions, standing up and walking toward the ball's location, are generated in the appropriate sequence. In general, this layer resolves the mechanical constraints or posture transitions, so that the upper layer does not need to consider the mechanical configurations.

The highest layer is a behavior generator, which generates a behavior based on both external events and the robot's internal states. For example, if a user gives a command such as "Move" to the robot, then this command event is applied to the highest layer. To simplify the process, the event is analyzed in this layer and generates the behavior associated with the command "Move."

The second architectural principle is applied to the individual robot parts such as the head, tail, and legs. As shown in Figure 5, for example, the head part has independent motivations such as keeping the head horizontal, tracking an object of visual stimuli, or tracking an object of sound stimuli. These behavioral motivations compete with each other. The implementation was to set fixed priorities for each motivation. In addition, commands from an upper branch have a higher priority. Consider that the quadruped robot has many postures, such as sitting, sleeping, and standing. In these postures, a designer has to create many motions for the head part. If the designer can create the motions for each part independently, it reduces the overall design time. Assume that if the head part has Nh kinds of motions, and the body part has Nb kinds of motions, then (Nh + Nb) designed motions make Nh × Nb combinations of motions. In addition, the movements of the head look natural because when the robot changes its posture from sitting to standing, the head remains horizontal, or when a user holds the body and swings it, the head part remains horizontal. But if the robot finds something to watch, the head part tracks the object. The tail part also has independent motivations, which express the emotional states of the robot through tail motion. It also includes keeping the tail horizontal. These kinds of motivations are fused by the agent architecture, which has an underlying tree structure.

It should be noted here that movements of each part are not totally free from each other. Mechanical interference is one of the main reasons. Dynamic balancing must be considered when movements of parts cause an unbalancing situation. If movements of a head part are designed with absolute joint angle sequences, the movements can be used only when a robot is in the same posture as the head movements are designed for. Therefore, it is better to design head movements with joint angle sequences relative to the gravity direction. Then, the designed head movements can be used when a robot is in any posture, such as sitting and standing. It is the same for tail movements. In our implementation on MUTANT, we designed about 10 movements for the head.

The third architectural aspect is the artificial emotional model. We evaluated sensor input with regard to the basic emotions of joy and anger and assigned appropriate dynamics to the basic emotions to configure this model. For instance, when joy is given a large value, the robot offers its paw if it sees a hand in front of it, but it refuses to offer its paw if anger is given a large value. In this way, different behavior was exhibited in response to the same stimulus, thus increasing complexity. When joy has an extremely high value, joy itself is expressed by movement or sound, such as laughter, and the same goes for anger, when the robot stamps the ground. Hence, emotions can be thought of as motivations for movement. A similar approach was taken for instincts, with dynamics assigned to virtual hunger and tiredness or to curiosity. The "hunger" is not real hunger, obviously, but is a simulation of hunger. Tiredness and curiosity are also simulated instincts. The robot makes a sound if hunger has a large value and rests if tiredness has a large value; if curiosity has a large value, it assumes search behavior (looking about restlessly), so that these can be seen as different motivations. The instincts and emotions model is described in the next section.

Figure 6 shows the agent architecture with its overall configuration. There are several perception modules: vision (color and obstacle) processing, sound processing, and posture processing, all shown on the left side of the figure. Detections of the location and size of stimuli in each perception module are performed. The results (locations and sizes) of perception are sent to the head part. The head part's architecture has three layers: attention, motion sequence generator (MoNet), and motor command generator (MCG). As shown in Figure 4, the target behavior generator for the highest layer actually provides "attention" for the head part. Since the target behavior of the head is basically to look at the object, we directly use "attention" for the highest layer of the head part. In addition, this process is very fast because the computation required is less expensive than "command recognition" or "target recognition" in the perception module.

The perception components further process and recognize things such as an orange ball, tone commands, and so on. These recognition results are sent to the body part and the instinct and emotional model component. In the body part, the target behavior generator that is implemented by finite state machines or automata in the highest layer (ATM) generates a behavior, the middle layer generates the action sequences (MoNet), and the lowest layer generates the motor command generator (MCG) as described above. The word MoNet is derived from Motion Network, which is the object name for our implementation of the action sequence model. The action sequences in the body part are then sent to the head part and suppress the independent motivations in the head part, as part of the mechanism of the tree structure architecture.

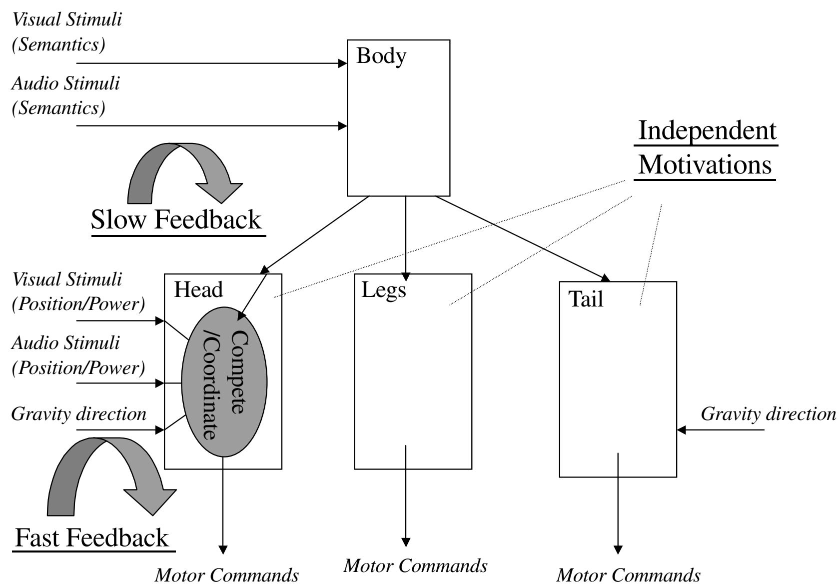

Fig. 5. Tree structure for agent architecture.

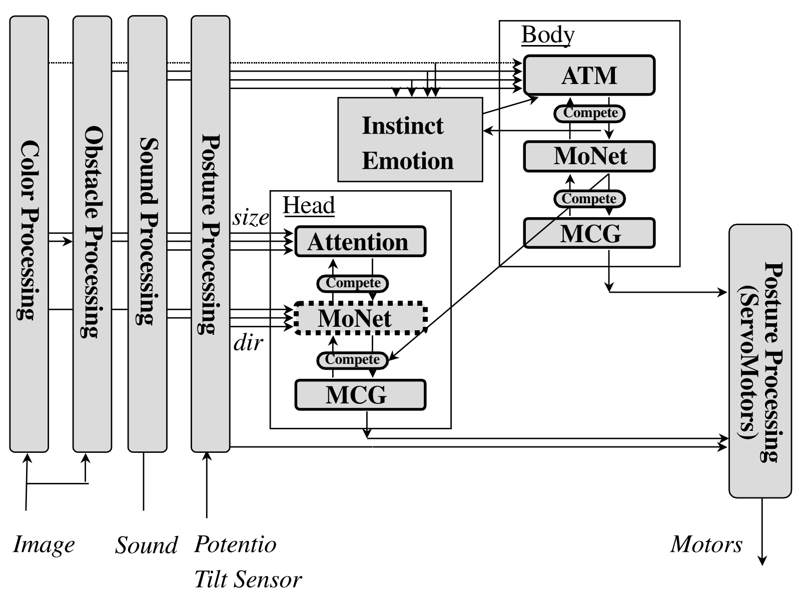

Fig. 6. The agent architecture of the prototype robot.

The prototype robot shown in the diagram is not equipped with a learning function. However, the robots (AIBO) we have recently announced have incorporated learning and development functions, which further expand the complexity of behavior and reactions. In the next section, we describe the agent architecture of AIBO with learning and growth functions.

3. Technologies Used in AIBO

In this section, we describe the technologies used in AIBO. First, we present how to maximize the complexity in AIBO by describing the agent architecture. One of the key issues related to the complexity is how to develop various motion patterns. We present the evolution of a walking pattern using a genetic algorithm, in which the robot itself can discover new walking gaits. We also present a description of the visual and sound processing used in AIBO. These technologies are very important for an autonomous robot to respond to external stimuli, which also make the robot lifelike.

3.1. Agent Architecture for AIBO

Starting from the agent architecture of MUTANT, we developed the agent architecture for AIBO. To increase the complexity of behaviors, we improved on the earlier agent architecture as follows:

1. *Behavior-based architecture:* As is the case for the behavior control architecture of MUTANT, we employ a behavior-based architecture for AIBO as well. For example, searching-tracking behavior is one of the behavior modules. Many different behavior modules are activated and selected by the action selection mechanism.

2. *Randomness:* Each behavior module consists of state-machines to realize a context-sensitive response. The state-machine is implemented as a stochastic state-machine, which enables the addition of randomness to action generation. For example, if there is a pink ball, the stochastic state-machine can determine that a kicking behavior is selected with probability 0.4, and a pushing behavior is selected with probability 0.6. Thus, different behaviors can be generated with the same stimuli, increasing the complexity of behaviors.

3. *Instincts/emotions:* This is the same idea as described in the previous section for the architecture of MUTANT. Simulating instincts and emotions generates motivations for behavior modules. The same stimuli can then generate different behaviors, again increasing the overall complexity of behavior. Of the numerous proposals put forward for emotions, we settled on six fundamental emotions based on the Ekman's model, which is often used in the study of facial expressions (Ekman and Friesen 1978). These are joy, sadness, anger, disgust, surprise, and fear. Just as with the instincts, these six values change their values according to equations, which are functions of external stimuli and instincts.

4. *Learning ability:* This feature is newly introduced for AIBO. Using the probabilities within the stochastic state-machine, we incorporate reinforcement learning in the architecture. For example, assume that when a hand is presented in front of the robot, there are several possible responses. Let's say, for example, there are five possible behaviors. One of the possible behaviors is the "give me a paw" behavior. At the beginning of learning, the probability for each possible behavior being manifested is 0.2. When the "give me a paw" behavior is selected with its initial probability, then the user gives a reward such as petting the robot's head. This causes an increase in the probability of the behavior from 0.2 to 0.4, and the other behaviors' probabilities decrease to 0.15. Then, if the hand is presented again in front of the robot, now the "give me a paw" behavior has a higher probability of being selected. Thus, a user can customize AIBO's response through reinforcement learning. This also increases the complexity of behaviors.

5. *Development:* This is also newly developed for AIBO. This learning ability involves long-term adaptation through interaction with users. Development can be considered as a slow changing of the robot's behavioral tendencies. Because we implement a behavior using a stochastic state-machine, which can be represented by a graph-structure with probabilities, we can change the graph-structure itself, so that completely different responses can be realized and a series of discontinuous changes can be observed during the robot's development over its lifetime.

6. *Various motions:* Finally, we implemented many motions, sound patterns, and LED patterns. This simply increases the complexity of behaviors.

3.2. Walking Pattern Generation

As described in the previous section, one of our strategies to increase complexity is to implement various kinds of motions, including walking patterns. AIBO ERS-110 has several walking patterns such as a slow, but steady, crawl gait pattern and a fast, but unstable, trot gait pattern. These patterns are used in different situations so that a user feels the developmental growth of AIBO. Most of the walking gait patterns are manually selected, but some of them are generated by the genetic algorithm. We developed a fully embodied evolutionary method for walking pattern generation (Hornby et al. 1999).

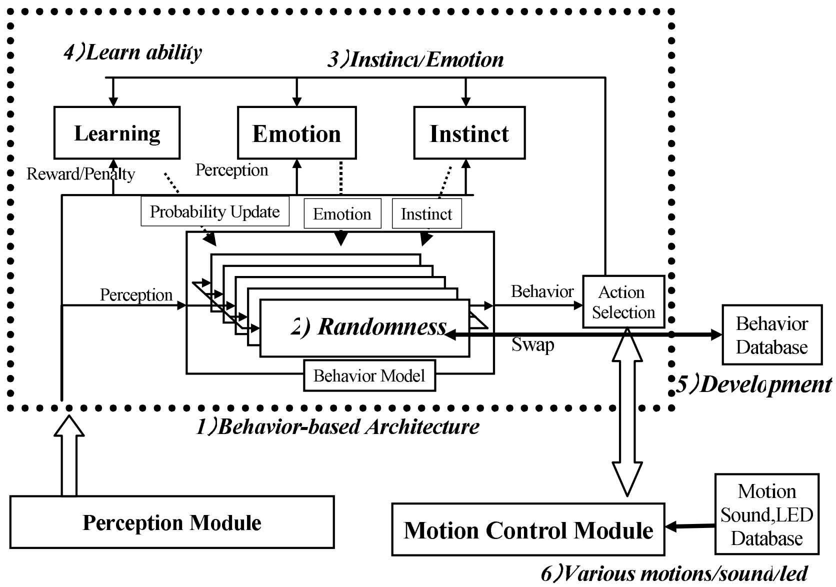

Fig. 7. Agent architecture of AIBO.

Namely, the robot, except for its power supply, can discover a proper set of parameters for the walking pattern generation program using only its computer and sensors. Because the experiment continues for more than 24 hours, it is impossible to use only batteries without human help. Therefore, we used an external power supply so that the experiment does not require a human assistant to perform.

A summary of key features of the genetic algorithm (GA) we used in the experiment is provided below. For more details, please refer to Hornby et al. (1999) and Fujita, Hornby, and Takamura (2000).

- A steady-state GA, which keeps fixed numbers of individuals (20 individuals in our experiments), is used for the GA algorithm.
- Real-value encoding is used for the genotype.
- Tournament selection strategy is used, in which we select some number of individuals (3 in our experiments) randomly from a group, keep 2 individuals that have the two highest values of the fitness function, and replace others by their children.
- For the fitness function, we use both speed and straightness of walking as the measures of quality.

Figure 8 shows the setup for this experiment. The area is about 1 by 2 meters and surrounded by 30 cm height walls. There are colored paper strips on the wall. The robot, using a color camera and position-sensing device (PSD), tries to walk toward the paper strip and evaluate how far and how straight it walks, using these sensors. Starting from a set of random initial values for the parameters, after about 20 generations, we can successfully evolve fast and stable walking patterns, which are about 6.5 m/min for a trot-gait pattern and 10.2 m/min for a pace-gait pattern. Figure 9 shows the pace-gait pattern acquired by the embodied evolution method.

3.3. Image Processing

Naturally, rich interaction with a human must be realized for a lifelike robot. From this point of view, a vision sensor is a key device for the autonomous robot. To make a robot small in size and weight and to reduce cost, we developed a micro-camera unit (MCU) using multichip technology, which is shown in Figure 10.

In addition, we developed a dedicated large-scale integrated (LSI) circuit, including a color detection engine (CDE), so that a robot can easily identify a colored object. Figure 11 shows how an input image is processed with the CDE. Each pixel in the input color image is compared with some threshold parameters to determine if it lies within the particular specified region. The result of this comparison is stored in a 1-bit image plane. If the thresholded value is in this region, the corresponding bit is set to 1; if not, it is set to 0. The CDE can detect eight different colors specified using a particular set of parameters.

We also implemented a multiresolution image filter bank, which generates three different images whose resolutions are 240×120, 120×60, and 60×30 pixels, respectively. Figure 12 shows these filter bank images. For example, a lower resolution image is used for color object tracking because

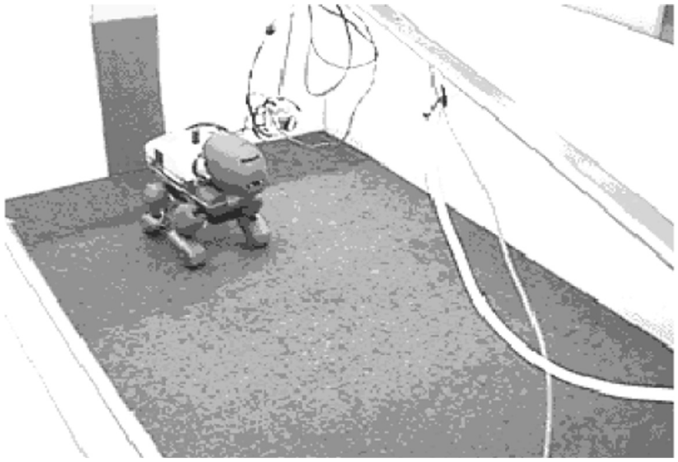

Fig. 8. The experimental setup of the embodied evolution for the walking pattern generation.

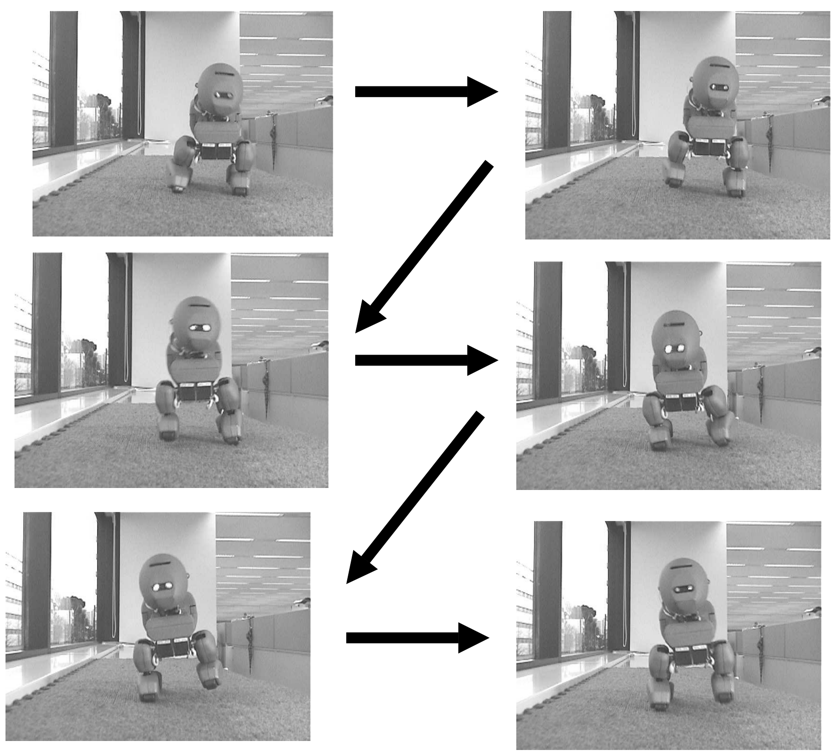

Fig. 9. Pace-gait pattern acquired by embodied evolution.

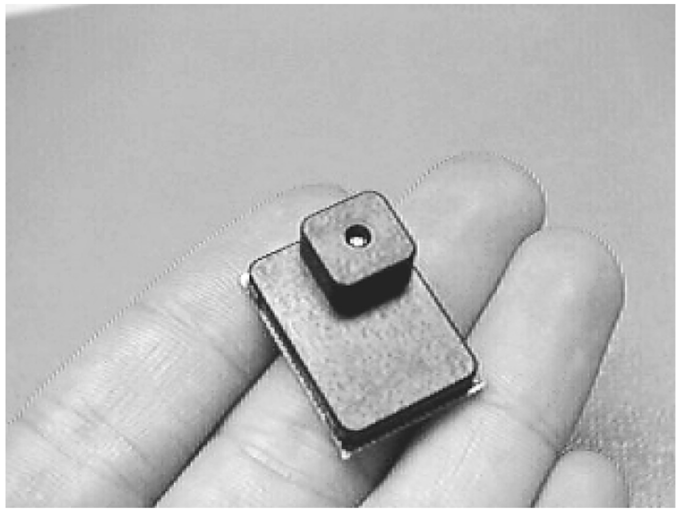

Fig. 10. A micro-camera module using multichip technology.

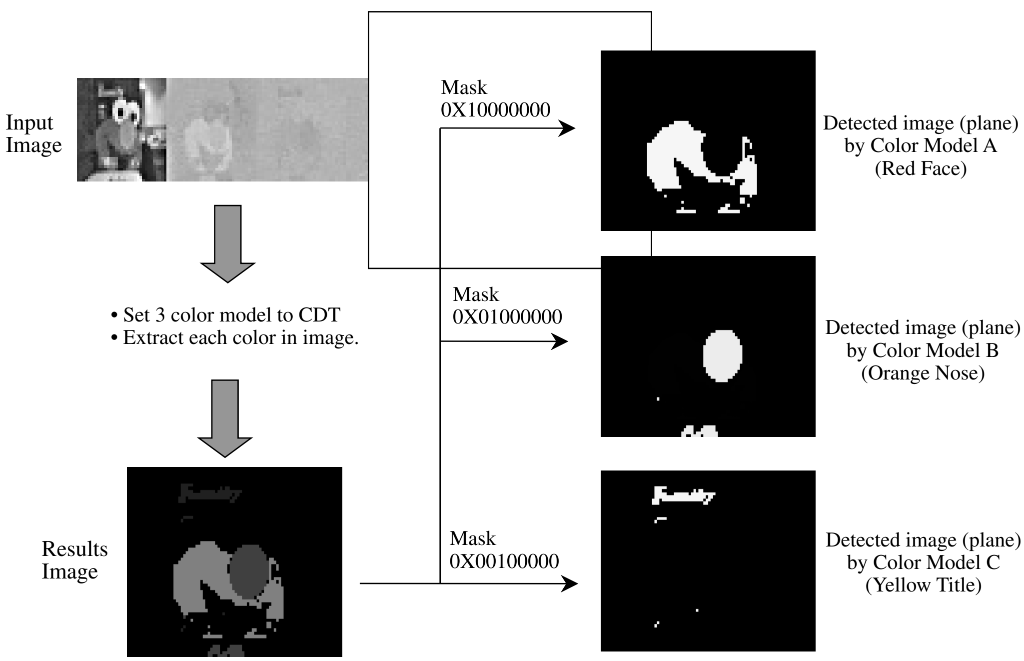

Fig. 11. Result of the color detection engine.

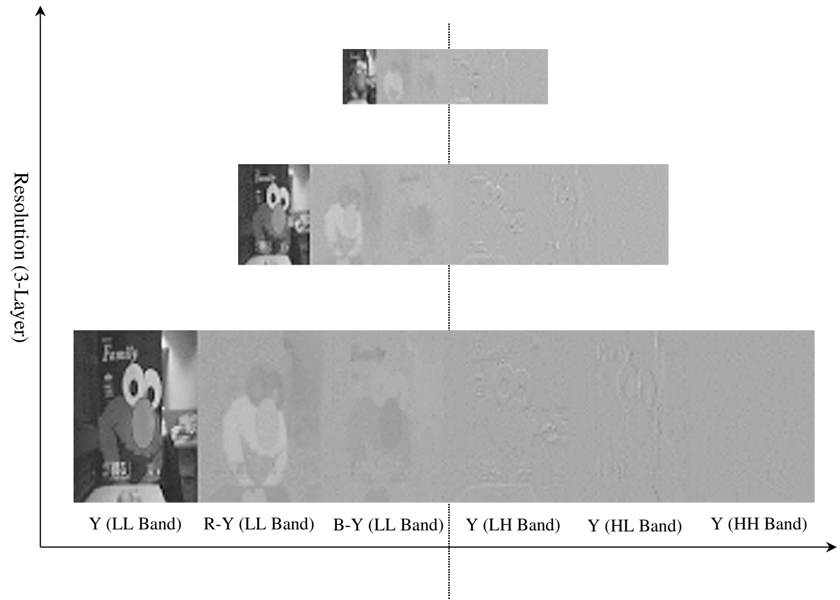

Fig. 12. A filter bank.

it must be processed as soon as possible so that fast visual feedback can be achieved. On the other hand, a high-resolution image may be used for pattern recognition because it requires the details found in the image.

3.4. Sound Processing

Interaction with environmental sound and speech is another important sensor channel for a lifelike robot. Again, the problem is how to realize robustness in the real world. Confounding factors include the following:

Noise: In an ordinary room or office environment, there are many noise sources such as air conditioning. Complicating things even further, the robot itself generates noise when it moves.

Voice interference: There is also voice interference generated by other people in the room. For example, when we demonstrate our robot, people often talk to each other around the robot.

To solve these two problems, we employ a "tonal language," which consists of tone signals with chords in arpeggio. As shown in Figure 13, a tone signal forms a line structure in the time-frequency graph. By using a "moving average" filter, it is easy to reject both noise and voice interference (Fujita and Kitano 1998).

4. Results and Discussion

In this section, we present our results gathered to date. While more rigorous evaluation would be desirable, it is unavailable as of the writing of this paper. As a result of our efforts to maximize complexity, the pet-type robot was clearly able to produce a big impact on an audience when we gave demonstrations. The most attractive behavior seems to be a recovery motion after the robot falls down. AIBO is able to get up from any posture, and sometimes it falls down when it kicks the colored ball. Conventional robots try to avoid falling down, but at least for a pet-type robot, falling down is a natural phenomenon, and recovering from a fall gives a lifelike feeling of the robot to the audience.

It is important for a robot having lifelike characteristics to react to stimuli and not only display complex motions. In addition, nonrepetitive reactions (i.e., to not react in the same way when the same stimuli are applied each time) are important to avoid boring the audience.

It is interesting to consider if users feel that the robot possesses "emotions." In fact, many users said, for example, "My robot is shy. He might feel fear now." Although we implemented artificial emotions, such users' explanations surprisingly often happened when the robot was not in the "fear" state. Thus, users tend to put explanations not related to the actual robot status but rather to the overall situation, regardless of the response of the robot.

This could be explained by Garfinkel's (1987) experiment. Assume that there is a system behind a curtain. Participants were told to ask questions that could be answered by yes or no. In addition, they were told to note why the system's answer was yes or no. Sometimes, the system's answers were controversial relative to the previous answers. However, the participants tried to explain the reasons, and after the experiment, most of them felt that there must be intelligence in the system. But behind the curtain, someone actually flipped

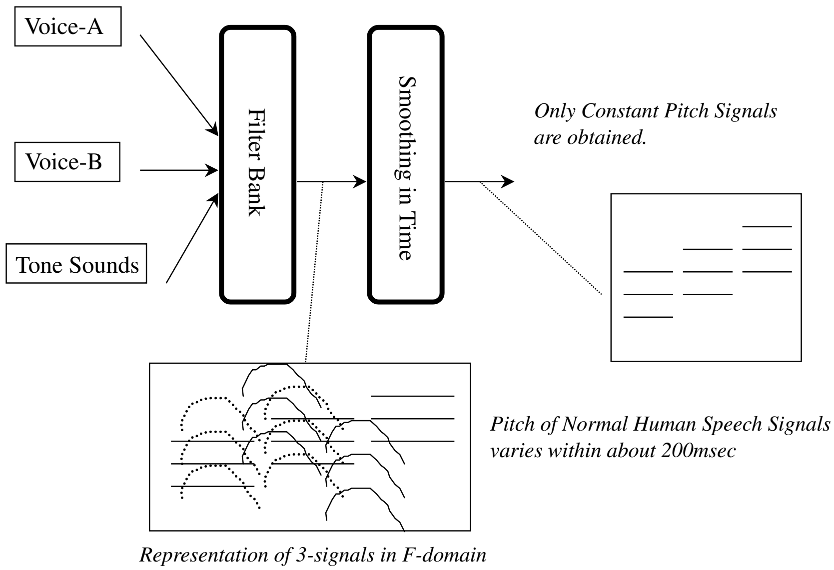

Fig. 13. Time-frequency characteristics of tones with voice interference.

a coin: if it was heads, the answer given was yes; if it was tails, the answer given was no. Thus, people tend to associate some intelligence or emotions with random phenomena when the situation is carefully designed. In our case, users tried to explain why AIBO reacts in certain ways, for example, by saying that "my AIBO must be shy, so he runs away" and so on.

5. Related Work

Much previous research has influenced the basis of the architectures described in this paper. The layered architecture shown in Figure 4 is a hybrid deliberative-reactive architecture, whose precursors are described in Arkin (1998).

The tree structure architecture shown in Figure 5 is a new concept. Several articles describe behavior-control architectures for a quadruped or a multilegged robot. The idea of the tree-structure architecture is similar to a multiagent system, however. Namely, each part can be considered as an agent that has its own separate behavioral strategy. The tree-structure architecture considers that many agents compete against and cooperate with each other. In addition, as our application is entertainment and not task-oriented robotics, and we focus on increasing the complexity of the robot's overall behavior and not on completing a specific task, it is possible for us to develop and use the tree-structure architecture.

Regarding reinforcement learning, the form implemented in AIBO is somewhat different from the ordinary reinforcement learning, as described in textbooks (e.g., Sutton and Barto 1998). Usually a reinforcement learning state is a categorized region in a perceptual space. Namely, if a robot executes a behavior, then its perceived world changes, which can be considered as a particular region in a perceptual space. This region forms the state used in ordinary reinforcement learning. However, in our model, our state is considered as a context, or a behavior state in a formal method of behavior control (Arkin 1998). Each state has "if" clauses, which check the situation of a perceptual world. It has also "then execute" clauses, each of which has an associated probability. Based on the probability, the system chooses one "then execute" clause and evaluates the resulting reward, which forms the basis for reinforcement learning.

The reason we use state-machine reinforcement learning is that a designer can easily control expected responses to the stimuli, as these probabilities are explicitly represented. The drawback of this method is that there is no chance for a new behavior to emerge.

Regarding the use of emotions, Blumberg (1996) and Breazeal (2000) are examples of related work. Blumberg implemented a virtual dog in a computer-generated virtual world. Blumberg used an ethological model to realize autonomous behaviors of the virtual dog, which is similar to a behavior-based architecture, but it fuses the external stimuli and internal states naturally. The main difference of our model from this approach is that Blumberg's model is a virtual dog, whose perception is far different from that of the real world. Our robot can misrecognize external stimuli, as it is difficult to recognize many objects in natural real-world settings. In addition, introducing a layered architecture and the tree-structure architecture is different from Blumberg's model. In his model, a tree-structure architecture is introduced, but it is a tree structure to categorize behavior classes as in ethological studies. Our tree structure is to form a multiagent system with robot parts, so that the complexity of behaviors increases.

Breazeal (2000) developed a talking head–type robot named Kismet. Her research focused on social interaction using detections of emotional signals of a human face and a human voice. The behaviors are basically emotional reactions, producing vocal sounds and facial expressions of the robot. The believability of the emotive expression of this robot is quite high so that people become actively engaged in interacting with Kismet. Kismet uses homeostatically regulated internal states, so that natural action selection can be performed, which is similar to our use of the instinct model. The advantage of Kismet is in detecting emotional signals of a human, which enables Kismet to return appropriate emotional responses, resulting in rich and engaging interaction with people.

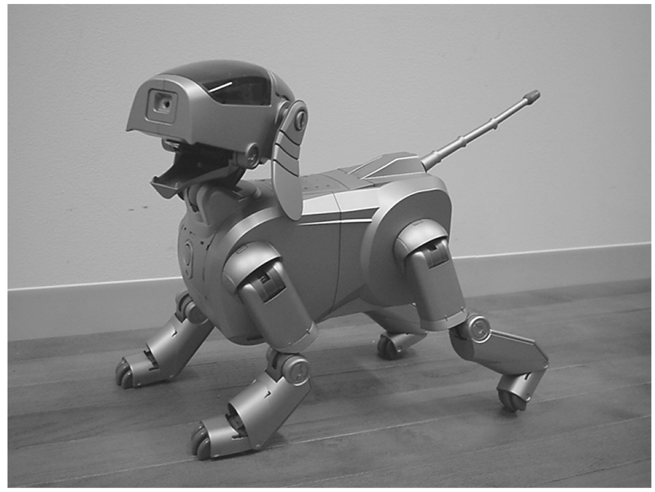

Fig. 14. A digital creature AIBO ERS-110.

6. Conclusion

In May 1999, we launched limited sales of our entertainment robot. This robot, called AIBO, is a product created on the basis of the results of studying a series of pet robot prototypes. The impact of AIBO can be assessed by the following facts. We announced that 3000 AIBO robots (ERS-110) would be manufactured for the Japanese market and 2000 for the United States, with a price tag of U.S.$2,500. We started to take orders only through the Internet. All 3000 AIBOs for Japan were sold within 20 minutes, and the 2000 robots for the United States were sold within 4 days. In the fall of 1999, we again made an announcement that 10,000 AIBOs (ERS-111) would be being manufactured for sale for Japan, the United States, and Europe. More than 130,000 requests came from all over the world. In this promotion, 45,000 AIBO robots (ERS-110 and ERS-111) were sold in the world overall. About 40,000 have been sold in Japan, where about 70% of our customers are males from 30 to 40 years old.

In 2000, we announced the second generation of AIBO (ERS-210), which sells for U.S.$1,500. As of April 2001, more than 50,000 robots have been sold, with about 80% in Japan. The main users are again 30- to 40-year-old males.

This type of entertainment application has served to accelerate research and development into autonomous robots. We hope that it will contribute to the understanding of not only the technological aspects of recognition and control but also the coexistence of people and robots. The problem of how to give an impression that a robot is alive is at the very heart of pet-type robots. To simplify the problem, we have confined our discussion to how to go about achieving complex movements, responses, and behavior of autonomous robots. To build a pet-type robot, we can learn from real animals and other living creatures. We can directly apply what we learn from them with regard to how they function and how they are built, as well as what significance their functions and structure have. Although we brought out our autonomous robot in a limited edition, we consider this to be the first step for the robot entertainment industry.

Acknowledgment

The author would like to thank Dr. Doi, the director of the Digital Creatures Laboratory, Sony, who provided the foundation of our research direction and has been supporting our research activities. The author also would like to thank the members of the Digital Creatures Laboratory and Entertainment Robot Company, Sony, who have been actually developing and building the robots. The author would also like to thank Professor Arkin at the Georgia Institute of Technology for assisting in English revisions and technical discussions to clarify the paper.

References

Arkin, R. 1998. *Behavior-Based Architecture*. Cambridge, MA: MIT Press.

Blumberg, B. 1996. Old tricks, new dogs: Ethology and interactive creatures. Ph.D. thesis, MIT.

Breazeal, C. 2000. Social machine: Expressive social exchange between human and robots. Ph.D. thesis, MIT.

Ekman, P., and Friesen, W. V. 1978. *The Facial Action Coding System*. Englewood Cliffs, NJ: Consulting Psychologists Press.

Fujita, M., and Kageyama, K. 1997. An open architecture for robot entertainment. *Proceedings of the 1st International Conference on Autonomous Agents*, pp. 435–440.

Fujita, M., and Kitano, H. 1998. Development of an autonomous quadruped robot for robot entertainment. *Autonomous Robots* 5:7–18.

Fujita, M., Kitano, H., and Kageyama, K., 1998. Reconfigurable physical agents. *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 54–61.

Fujita, M., Hornby, G., and Takamura, S., 2000. Automatic evolution of gaits of quadruped pet-type robot with genetic algorithm. In *Genetic Algorithm 4*, ed. H. Kitano. Sangyo-Tosho.

Garfinkel, H. 1987. *Studies in Ethnomethodology*. Cambridge, UK: Blackwell.

Haliday, T. R., and Slater P. J., eds. 1983. *Animal Behavior*. Cambridge, UK: Blackwell.

Hornby, G., Fujita, M., Takamura, S., Yamamoto, T., and Hanagata, O., 1999. Autonomous evolution of gaits with the Sony quadruped robot. *Proceedings of Genetic and Evolutionary Computing Conference (GECCO)*, pp. 1297–1304.

Shibata, T., Yoshida, M., and Yamato, J. 1997. Artificial emotional creature for human-machine interaction. *Proceedings of the IEEE System, Man, and Cybernetics*, pp. 2269–2274.

Sutton, R., and Barto, A. G. 1998. *Reinforcement Learning*. Cambridge, MA: MIT Press.

Velasquez, J., Kitano, H., and Fujita, M. 1998. Open architecture for emotion and behavior control of autonomous agents. *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 473–474.