Evolving Behavior Sequences for a Humanoid Entertainment Robot

Wei-Po Lee Jih-Shiou Jong Tsung-Hsien Yang

Department of Information Management,
National Sun Yat-sen University, Taiwan

Abstract: One of the most important issues in developing entertainment robot is human-robot interaction, in which the robot is expected to learn new behaviors specified by the user. In this paper we present an imitation-based mechanism to support robot learning and use evolutionary computing to learn new behavior sequences. We also propose several advanced techniques at the task level and the computational level to evolve complex sequences. To evaluate our approach, we use it to evolve different behaviors for a humanoid robot. The results show the promise of our approach.

Keywords: Entertainment Robot, Humanoid Robot, programming-by- demonstration, Genetic Algorithm

I. INTRODUCTION

In recent years, entertainment robots have been considered the main trend of the next-generation electronic toys, and this type of robots has become an important application of intelligent autonomous robot [1][2]. Building fully autonomous artificial creatures with human-like intelligence is a long-term goal and it has not yet achieved at the present stage. However, with current technologies in computing and electronics, and new knowledge in ethology, neuroscience and cognition, it is now possible to create embodied prototypes of artificial living toys acting in this physical world.

One of the most important issues in developing entertainment robot is human-robot interaction, in which the robot is expected to learn new behaviors specified by the user. Entertainment robots are service robots to live with people, therefore we need to furthermore apply knowledge of biology and ethology to derive design principles for interaction and learning. Imitation is a powerful mechanism in social animals for learning and delivering new knowledge [3][4], and some researchers have proposed to endow a robot with such social-cognitive skills [5][6][7]. In this work, we choose to implement such a mechanism to provide a special kind of human-robot interaction.

In this paper we present an evolution-based approach to realize the imitation-based learning. In our work, the robot is shown how to perform the desired behavior first, and during the period of human demonstration, the behavior sequences are recorded and analyzed. Then the Genetic Algorithm is employed to evolve the behavior sequences: to determine how to rotate different motors on the robot's body parts to produce the same behaviors. To evaluate our approach, we use it to evolve different behaviors for a humanoid robot. As for complex behavior sequences, we propose some advanced techniques for performance enhancement, including the division of the overall behavior at the task level, and the exploitation of priori information and exploration of search space with an adaptive mutation at the computational level. The preliminary results and analyses show that our approach can successfully and effectively evolve behavior sequences for the robot.

II. EVOLVING BEHAVIOR SEQUENCES

1. System Overview

As mentioned above, this work aims to establish an imitation-based learning framework to evolve behavior sequences for a humanoid robot. Fig. 1 illustrates such a framework that mainly includes two parts for active learning and passive learning, respectively. The active part involves a gesture recognition procedure, which is to capture and record images of the demonstrator, and then extract the curve for further interpretation. If a match is found, the corresponding behavior module is retrieved and activated from the behavior library. The control code for generating the sequence is then sent to the robot. The library can be pre-constructed to include some behavior primitives as basic components, and is gradually expanded by the user during the human-robot interaction procedure.

The passive part is to build a new behavior that does not exist in the behavior library. Here the new behavior means a completely new sequence (if no sequences in the library matched any part of the newly demonstrated one) or a sub-sequence obtained from the demonstrated sequence (after the partially matched parts have been removed). Because this work focuses on the passive part of the demonstration-based learning framework, therefore we will concentrate on describing the development of our learning mechanism, in which a modified genetic algorithm is employed to evolve behavior sequences.

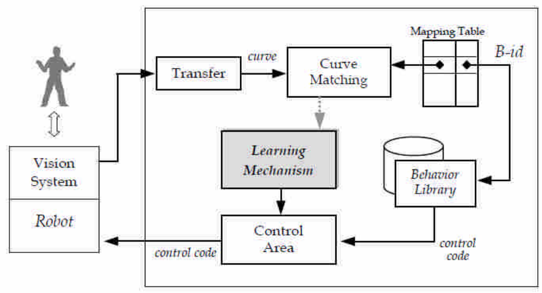

Fig. 1. The system overview.

2. Evolving Behavior Sequences

The first important step in evolutionary computation is to choose a proper representation for the individual. The goal here is to drive the motors of a humanoid robot to produce the same behavior sequences as that of the human demonstrator. We thus use a direct encoding scheme that takes the motors to be considered and their corresponding activities as gene pairs to constitute a chromosome. That is, a chromosome is a fixed length string that records a sequence of motor activities along time domain. The odd genes indicate the motor identifiers and the even genes are the motor rotating angles. To provide the robot a reasonable number of time steps to produce the target behavior sequence, the length of the chromosome is defined as the ratio of the estimated displacement of the behavior trajectory to the minimal motor speed.

As the behavior shown by the demonstrator may involve different body parts (for example, both hands in this work), the motors drive these parts can thus rotate independently at the same time. In such situations, the behavior trajectories of different parts can be built separately, in which only the motors driving the same part are used in the corresponding chromosome. However, in the above situations the synchronization problem between different body parts needs to be considered. For example, in Fig. 2, the demonstrator moved his right hand but halted the left hand (the *x*, *y*, *z* coordinates remain the same) in two short periods T_1 and T_2. Therefore, when the behavior sequences for the two hands are evolved separately, some motor genes with zero rotating angles (i.e., halting the motors) need to be inserted to the solution for the left hand to keep it synchronized with the right hand.

From the computational point of view, using motor genes with zero rotating angles will generate some redundant gene pairs and introduce extra difficulty in the evolutionary process, so we employ an alternative way to deal with the synchronization problem instead. In our method, sequences for different body parts are aligned and the halting periods are used to divide these sequences. For example, in Fig. 2 the two sequences for the right and the left hands are divided into four independent sub-sequences. Then the solutions are evolved separately for each sub-sequence and combined as the overall solution.

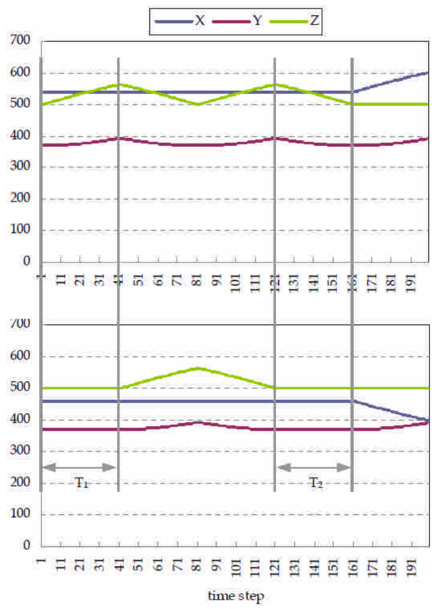

Fig. 2. An example of behavior sequence in which the upper (lower) part shows the *x*, *y*, and *z* coordinates of a tracking mark on the demonstrator's right (left) hand.

After the genetic representation has been defined, the next step is to evaluate individuals to determine their fitness for the creation of a new population. Here, the goal is to measure how the behavior sequence produced by the individual is close to the original sequence shown by the demonstrator. It is to accumulate the deviations derived from the motion effect of the motor activities described in a chromosome. If the target and actual positions for a tracking mark at time step t are $p(t)$ and $p'(t)$, respectively, the closeness of the target and actual sequences can be measured as:

$$closeness = \sum_{t=1}^{T} c(t) \quad \text{and}$$
$$c(t) = 1 \text{ if } d(t) < d_{thread}\,; \quad c(t) = 0 \text{ otherwise}$$

In the above equation, T is the number of time steps of the robot's performing the task; $d(t)$ means the distance between the target and actual positions at time t; and $c(t)$

indicates whether the actual sequence can approximate the target sequence at time *t*.

As the *closeness* measures the position difference of two sequences at discrete time points, it is thus possible to obtain two sequences with very similar closeness but have different motion trajectories. To distinguish among sequences precisely, a penalty term is introduced to estimate the differences of motion trajectories of the target and actual sequences as:

$$\sum_{1}^{T}[v_p(t)-v_p(t-1)]-\sum_{1}^{T}[v_{p'}(t)-v_{p'}(t-1)]$$

in which v is the velocity of the tracking mark (i.e., $v_p = |p(t) - p(t+1)|$ and $v_{p'} = |p'(t) - p'(t+1)|$). With the above two kinds of measurement, we then define the fitness function as their weighted sum:

$$fitness = closeness - \alpha \times penalty$$

Based on the above fitness function, the individuals can be evaluated and their performance can be determined. Then a certain selection scheme is used to choose parent individuals. In our implementation, the *tournament* selection scheme is employed to choose parents. Then, three genetic operators, *reproduction*, *crossover*, and *mutation*, are applied to create a new population for the next generation. As our representation includes two types of data (motor identifier and rotating angle), two kinds of crossover are used for the recombination of individuals: uniform crossover for motor genes and arithmetical crossover for rotating angles.

3 Evolving complex behavior sequences

After describing how an evolutionary framework is developed to learn new behavior sequences, this section presents how it is furthermore extended to evolve complex sequences from different points of view. Here the complex sequence means a sequence that needs specific skills to achieve; it is not necessarily a long sequence. The first way to evolve complex sequences is to reduce the complexity from the task level. That is, to take a divide-and-conquer technique to decompose a complex behavior sequence into several sub-sequences, evolve solutions for the sub-sequences and then combine them together for the complete sequence. Though this method provides a direct way to reduce task complexity by the division of original sequence, however, its corresponding disadvantages need to be seriously considered. For example, it is tedious for the user to deal with the large amount of tiny subsequences. Also it becomes difficult to attach suitable semantic interpretations for the tiny sequences and this is important to many service robot applications.

The other direction to encounter the complexity problem is to take a computational perspective to enhance the searching performance. Two methods have been developed in this work. The first method involves the exploitation of the priori knowledge in searching solution space. It is to retrieve some behaviors sequences, which are similar to but different from the target behavior, from the behavior library to the initial population. It is expected that with the guidance of these relatively good solutions, the evolutionary search can become more efficient and the target behavior can thus be obtained.

The other method is to develop an adaptive mutation scheme to maintain population diversity during the evolutionary process. It is to calculate the standard deviation of the fitness of all individuals at each generation (i.e., sd_i), and to record the maximum of the standard deviation from the first generation (i.e., *max-sd-so-far*). Then the ratio of current standard deviation to the maximum value is used to determine the mutation rate, which is the default mutation rate multiplies a factor of ($1 - sd_i/max\text{-}sd\text{-}so\text{-}far$). In this way, the population is guaranteed to include certain new individuals to main its diversity.

III. EXPERIMENT RESULTS

To assess the proposed methods, we used them to evolve different complex behavior sequences. In this work, we developed a simulator to generate the behavior sequences to be achieved and record the coordinates accordingly, to save the effort of human demonstration. In addition, to fit the hardware restrictions of the real robot used in this work, three motors, on the shoulder (pitching), elbow (rolling), and twist (rolling) joints, were used to drive each hand. Here, each motor was allowed to rotate within the range of +90 to –90 degrees. The behaviors evolved from simulation were transferred to a real humanoid robot for verification.

In the experiments, we pre-compiled three complex behavior sequences, as illustrated in Fig. 3 (due to the space limitation, only one behavior is shown), to evaluate the methods mentioned in section II.3 in evolving complex sequences. We also conducted a set of experiments that took the original framework to evolve the same tasks for performance comparison. The results without advanced techniques are shown in Fig. 4, in which no successful runs can be obtained.

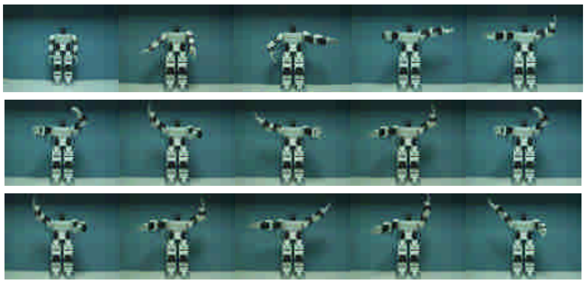

Fig. 3. One of the target behaviors.

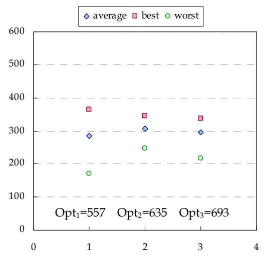

Fig. 4. The results for evolving behaviors with different population size (The *y*-axis means fitness value, and the indices 1, 2, and 3 in the *x*-axis represent experiments for three different behaviors, respectively).

1. Task decomposition

As described in section II.3, we can adopt the divide-and-conquer method to reduce task complexity, evolve partial solutions for subsequences, and then combine the partial solutions to obtain the final solution. To evaluate the performance of task decomposition, we divided the each of the tasks into two behavior subsequences, and then four subsequences (with approximately the same length), to examine the corresponding effect. In the experiments, ten independent runs were conducted for each of the divided subsequences. The population size is 1000, the crossover rate is 0.7 and the mutation is 0.003.

Fig. 5 lists the results of dividing the original tasks into different number of subsequences. As can be seen, initially all of the three complex behavior sequences cannot be evolved correctly. After each of the sequences was divided into two subsequences, the first behavior can be evolved successfully four times in the ten runs, and the third behavior, one time. We then furthermore divided them again into four subsequences and repeated the evolutionary runs again. As can be expected, the target tasks became more achievable after the division was performed again. But it should be noted that more and more tedious experimental runs will be needed if the task are divided iteratively. Also it will become difficult to attach semantic labels for the tiny sub-behaviors to be reused in other applications.

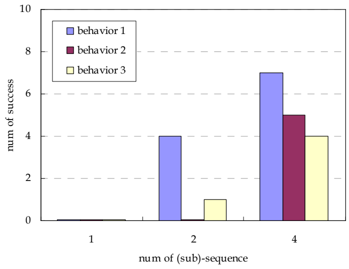

Fig. 5. The number of successful runs in evolving complex sequences by task decomposition.

2. Advanced evolutionary techniques

In addition to reducing complexity from the task level to evolve complex behaviors, we also investigate whether the priori information of a specific problem can be used to derive solutions in a more efficient manner. Here, the priori information means the similar behavior sequences already recorded previously in the behavior library. In this work, ten simple behaviors are pre-defined as primitive robot behaviors. To exploit the information already obtained, we measured the similarity between the target behavior and the ones recorded in the library, and then chose and inserted three most similar behaviors to the initial population in the evolutionary process.

Ten independent runs have been conducted for each of the tree tasks. In each run, a population of 1000 individuals was used, in which three of them were taken from the library as described above. Fig. 6(a) presents the results. It shows that in average the experiments with priori information have better performance than those without using information. However, we also noticed that with the guidance of priori information, the runs converged more quickly than those without using information. This feature caused evolutionary runs to lose population diversity and became premature. Consequently, we cannot obtain successful solutions.

To prevent the above premature situation, we have developed an operator of adaptive mutation to maintain population diversity. It is to introduce randomness of the population based on the standard deviation of the fitness of all individuals. In this set of experiments, the mutation rate of each generation was determined by the criteria described in section II.3. To observe the effect of adaptive mutation alone, we conducted ten runs for each of the three complex behaviors in which all other parameters were the same as the abovementioned methods. Fig. 6(b) summarizes the results. As can be seen, better results can be obtained, though there is no successful run obtained.

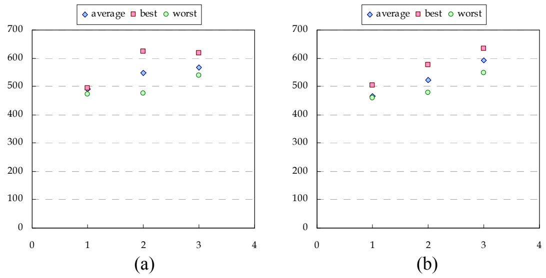

Fig. 6. The results of using priori information (a) and adaptive mutation (b) to evolve robot behaviors.

From the above two sets of experiments, it can be seen that using priori information or adaptive mutation can both improve search performance and evolve better solutions, but no perfect solutions can be obtained from all of the runs. Therefore, following the above experiments, we combine both strategies to conduct evolutionary runs (with the same parameter settings as before) in evolving complex behaviors. Fig. 7 summarizes the results, in which the numbers of successful runs for the three behaviors are now 5, 2, and 3, respectively. Fig. 8 shows how the fitness curves converged during the typical runs (for the behavior illustrated in Fig. 3). As can be seen clearly, by integrating two useful strategies to exploit all their advantages, perfect solutions can be evolved. It shows the success of our approach.

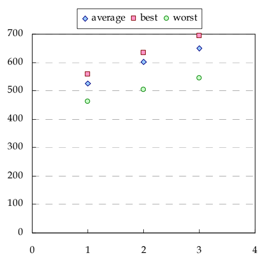

Fig. 7. The results of using both priori information and adaptive mutation.

IV. CONCLUSIONS

In this paper, we have indicated the importance of developing entertainment robots as an intelligent robot application. To realize the development of adaptive entertainment robot, an imitation-based learning mechanism has been constructed with which the user can teach the robot how to perform the expected behavior through demonstration. Here we use an evolutionary approach to support the imitation-based robot learning, and implement several advanced techniques at the task level and the computational level to evolve complex behavior sequences. The experimental results show that by the exploitation of priori information and the exploration of solution space with an adaptive mutation, the complex behavior sequences can be evolved successfully.

Based on the presented work, we are currently extending our approach to evolve robot behaviors with coordination of both hands and legs. Since this type of behaviors involves the balance of robot body, it becomes more difficult to achieve. We are investigating more advanced computational techniques to deal with the relevant problems. In addition, we plan to integrate an efficient vision system into the robot to acquire and analyze the demonstrator's behavior sequences in real time.

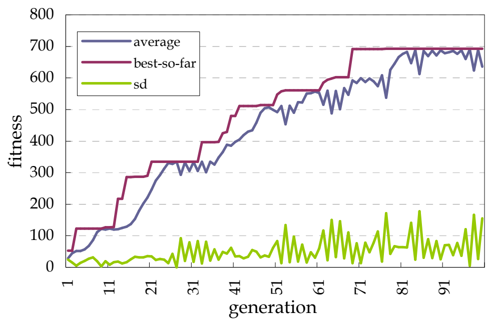

Fig. 8. The fitness curves of the typical run in evolving behaviors by both priori information and adaptive mutation.

REFERENCES

[1] Veloso MM (2002), Entertainment robotics. *Communication of the ACM*, 45, 59-63
[2] Fujita M (2004), On activating human communications with pet-type robot AIBO. *Proceedings of the IEEE*, 92, 1804-1813
[3] Rizzolatti G and Craighero L (2004), The mirror-neuron system. *Annual Review of Neuroscience* 27, 169-192
[4] Zentall TR (2006), Imitation: definitions, evidence, and mechanisms. *Animal Cognition* 9, 335-353
[5] Dantenhahn K (2007), Socially intelligent robots: dimensions of human-robot interaction. *Philosophical Trans. of the Royal Society B*, 362, 679-704
[6] Breazeal C (2002), *Designing Sociable Robots*, MIT Press, Bradford Book.