Interactive Behavioral Design between Autonomous Behavioral Criteria Learning System and Human

Min An*, Toshiharu Taura
Graduate School of Science and Technology, Kobe University,
Tel/Fax: (078) 803-6481
E-mail: anwang@mi-5.scitec.kobe-u.ac.jp
1-1, Rokkodai, Nada Kobe, 657-8501, Japan

Abstract
In this study, we suggest a learning system for the autonomous creation of novel behavioral patterns through interaction with human beings. Two kinds of evolutionary calculation are used in the proposed system: genetic algorithms (GAs) with which the system learns behavioral patterns and genetic programming (GP) with which the system acquires behavioral criteria depending on the behavioral pattern learned with GAs. In addition, novel behaviors are designed with the learned behavioral criteria through interaction between the system and the user. Since the system generates new behaviors with new behavioral criteria instead of simply combining existing behaviors, various novel behaviors can be created. Computer experiments have been carried out to demonstrate the effectiveness of the proposed system.

Key words
robotics, genetic algorithm, genetic programming, behavioral design, interactive design

1. Introduction
Robots are being increasingly used for therapy, rehabilitation [1], and entertainment [2]. However, conventional robotic behaviors are currently decided a priori by programmers. Next-generation robots must be able to determine what they should do, how to do it, and how to design a suitable behavioral pattern. Some researchers have analyzed human action sequences and the imitation of human behavior [3]. Such research has solved the problem of how to learn behavior. Nonetheless, there has been little research focusing on teaching a robot to behave autonomously.

Recently evolutionary techniques like genetic algorithms (GAs) have attracted much attention as optimization methods. Living things have evolved by developing new abilities and new behaviors through such natural systems as selection, crossover, and mutation. We believe that similar mechanisms can also be applied to artificial things. In many cases when a new design is created, some improvement is made to past designs, or multiple elements may be combined. This can be regarded as evolution [4]. Using evolution simulation techniques, a computer is able to create new designs of behaviors.

Methods for the simulation of evolution for design have been studied. In such methods, fitness functions for other objects are difficult to define. With respect to this problem, interactive evolutionary computation (IEC) has been actively studied [5]. This approach is highly effective for problems in which it is difficult to define the fitness functions, such as in design, adjustment or creation based on human senses. As examples, IEC has been applied to the design of cars [6] and the design of women’s dresses [7].

In this study, we suggest a system including two kinds of evolutionary calculations, genetic algorithms (GAs) and genetic programming (GP). GP is used to learn a behavioral criterion and to construct new behavioral criteria, and GAs are used to determine how suitable each individual behavior is for the performance on the basis of the criteria constructed by GP. The characteristics of this system are its autonomous learning of basic behavioral criteria and recreating of a number of novel behavioral patterns through interaction between the learning system and human beings.

Two kinds of stages are used in the proposed system: stages where the system learns behavioral criteria and reproduces behavioral patterns autonomously by imitative techniques; and stages where novel behaviors are designed through interaction between the system and the user.

The remainder of this paper is organized as follows. In Section 2, the imitative techniques used in this study and behavioral criteria are defined. In Section 3, the proposed system is explained. Then, in Section 4, the results of computer experiments are shown.

2. Definitions
When a robot learns behavioral patterns, it may focus on the coordinates of the finger, the joint of the elbow, and the joint of the shoulder, and their angles; otherwise, it may only pay its attention to evaluation criteria such as ‘smoothly’, ‘quickly’ and so forth, which describe the whole movement from the start-point of the movement to the goal-point [8].
In this study, the evaluation criteria are defined as evaluation functions for establishing behaviors. They carry information on what the robot wants to do and evaluate whether it performs the behavior well. For example, if the robot wants to touch an apple, then the apple is the target of the behavior. One of the criteria is for evaluating whether the fingertip reaches the apple, as shown in equation 1.

\[ E_t = (X - x_T)^2 + (Y - y_T)^2 + (Z - z_T)^2 = 0 \]  

(1)

Here, \( T \) indicates the numbers of steps needed to reach the apple, \( x_T, y_T, \) and \( z_T \) are the coordinates of the fingertip of the robot, and \( X, Y, \) and \( Z \) are the coordinates of the apple.

Another kind of criteria is found in the behavior of touching the apple, and is for evaluating the characteristics of the trajectory between the start position of the fingertip and the apple, such as “Minimum Distance” as shown in equation 2.

\[ E_d = \int_0^{T-1} \{ (x_{t+1} - x_t)^2 + (y_{t+1} - y_t)^2 + (z_{t+1} - z_t)^2 \} dT \]  

(2)

3. The proposed design system

In a conventional imitation study, a robot only learns one behavioral pattern for a certain task from a person, and the learned behavioral pattern can only be adapted to very similar tasks. For example, if the robot imitates the behavioral pattern of touching a target, it can then only touch several different targets by following the same behavioral pattern. It can neither change the simple pattern, nor apply the learned pattern in tasks other than touching a target, such as dancing. However, in this study, we aim to produce a robot that can not only learn a model behavioral pattern but can also develop novel behavioral patterns on the basis of the learned behavioral pattern. Thereby, in stage 1 in our suggested framework, we let the robot acquire several basic behavioral criteria for evaluating the behaviors of a model person or a model robot, and then it reproduces the behaviors based on the learned basic behavioral criteria, autonomously.

In stage 2 of the framework, we suggest an interactive process with a user for designing novel robotic behavioral patterns. The behaviors, which may be novel robotic behavioral patterns, are autonomously reproduced by evolutionary calculations based on the learned basic behavioral criteria in our learning system. They are shown on a computer screen by the learning system, and a user can select two or more preferred behaviors from among them. Then, the robot combines the basic behavioral criteria mentioned above to produce a new behavioral criterion. Finally, the robot autonomously performs a new behavioral pattern based on the combined new behavioral criterion. It is worth noting that we do not allow the robot to simply change the coordinates of movement in the model behavior, because the changed behavior would probably be unnatural; moreover, the user would lose interest in the dancing performance of his robot if the robot could only change the coordinates of a fingertip or an elbow in a new dancing pattern.

The system is composed of two stages: stage 1 for learning two model behaviors with GA and constructing two behavioral criteria \( C_1 \) and \( C_3 \), and stage 2 for combining \( C_1 \) and \( C_3 \), and recreating behaviors depending on the combined criterion.  

3.1 Individuals of Genetic Algorithms

In our simulation, the behaviors are learned through GAs. The variation of each angle is presented as a GA gene, so that a series of variations from the start point to the target point is replaced by one individual in the GA. The elements from two two best solutions combine and mutate to produce a new individual, and thus the new individual will be evaluated. If the new individual is better than one of the old individuals, then it survives and the least effective one dies. This process continues through a number of iterations.

The three-dimensional, three-degree-of-freedom scalar manipulator shown in Figure 3 was used to perform the designed behavioral patterns in our simulation. The details of GA coding are shown in Figure 4. The three elements of the arm are named the upper arm, forearm and hand. In step \( t \), the coordinates of a fingertip, a wrist and an elbow are \((x_{\text{finger}}, y_{\text{finger}}), (x_{\text{wrist}}, y_{\text{wrist}})\) and \((x_{\text{elbow}}, y_{\text{elbow}})\), and the angles of motion are \( \theta_{\text{finger}} \), \( \theta_{\text{wrist}} \), and \( \theta_{\text{elbow}} \), respectively.

![Fig. 1 Coordinates and angles of a robotic arm](image)

![Fig. 2 GA coding](image)
3.2 Individuals obtained by GP

We propose a method of inferring behavioral criteria \[9\]. Initially, the imitating robot does not have any sample data for inferring a model’s behavioral criterion. Therefore, the initial generation of inferred behavioral criteria is randomly produced by GP. Then, the imitating robot learns the object’s behavioral patterns through GAs, in which the evaluation functions are the behavioral criteria randomly produced by GP. The learning efficiencies vary, because the learning process is based on different behavioral criteria. Therefore, the most adaptive behavioral criterion that reproduces the behavior most efficiently can be chosen as the behavioral criterion of the next generation. As the process is repeated until the final generation, a behavioral criterion that is the same or the nearly same as the model’s criterion can be acquired. The method of inferring behavioral criteria is explained in ref. \[9\], and interested readers are referred to that paper for details. In addition, a new method that evolves the design process though interaction between the user and the system in stage 2 is explained in subsection 3.3.

Sample data are taken from the trajectory learned from the previous generation (except that initial data were produced randomly). The strategy for generating sample data is shown in Figure 4. The coordinates of each three points are arranged as one sample set. For one GP, we select six sample sets from one trajectory. The three points are chosen randomly, because we do not know which combination is the best. The numbers of the maximum initial depth and the maximum evolution depth are set to 3 and 16. For example, we can select the following six sets of samples from Figure 4.

\[
\begin{align*}
SS1 &= \{p_1, p_2, p_3\} = \{1.0, 3.0, 1.0, 3.33, 2.5, 3.5\}; \\
SS2 &= \{p_3, p_4, p_5\} = \{2.5, 3.5, 3.0, 3.66, 4.0, 4.0\}; \\
SS3 &= \{p_5, p_6, p_7\} = \{4.0, 4.0, 4.5, 4.16, 5.0, 4.33\}; \\
SS4 &= \{p_7, p_8, p_9\} = \{5.0, 4.33, 5.8, 4.6, 6.5, 4.83\}; \\
SS5 &= \{p_1, p_3, p_5\} = \{1.0, 3.0, 2.5, 3.5, 4.0, 4.0\}; \\
SS6 &= \{p_5, p_7, p_9\} = \{4.0, 4.0, 5.0, 4.33, 6.5, 4.83\}.
\end{align*}
\]

3.3 Combination of behavioral criteria

In stage 2, the two behavioral criteria \(C_1\) and \(C_2\) constructed in stage 1 are combined into a new behavioral criterion. An element \(T\) is decided by the user that indicates the total time taken by the robot fingertip to move from a start point to a goal point. In addition to \(T\), \(T_c\) and \(P_{Tc}\) are also decided by the user. \(T_c\) indicates the number of times criteria are combined, and the \(P_{Tc}\) indicates the positions in which criteria are combined. The relationship between \(T_c\) and \(P_{Tc}\) satisfies the following:

\[
0 \leq T_c \leq 0.2T, \quad (3)
\]

\[
P_{Tc} = (P_i; i = 1, \ldots, Tc). \quad (4)
\]

\(T_c\) is limited less than 0.2\(T\), because behavioral patterns cannot be created effectively if the behavioral criteria are changed too frequently.

Here, \(T = 9\), \(T_c = 1\), and \(P_i = 3\).

Figure 5 illustrates the method of the combination of behavioral criteria. \(C_1\) and \(C_2\) are arranged in order from left to right. The former three \(C_i\)s instruct the system to generate the behavioral pattern between \(t_0\) and \(t_f\) according to \(C_i\), the latter six \(C_i\)s instruct the system to generate the behavioral pattern between \(t_{Tc}\) and \(t_0\) according to \(C_i\), and so on. Here, \(t_i\) indicates the \(i\)th movement, and \(t_f\) is the final movement.
Two robots are created using OpenGL as shown in Figure 6, called the “model robot” and the “imitation robot”. In this simulation, the “model robot” teaches the “imitation robot” basic behavioral patterns instead of a model person. In the future, we will let a person teach the basic behavioral patterns but not at present, because we do not have sufficient equipment for motion capture.

The arms of both robots are created to be three dimensional. Two kinds of computer experiments have been carried out. In experiment 1, the system aimed to design a movement in a workplace, and in experiment 2 the imitation robot attempted to perform a sports behavior, namely, pitching a baseball.

### 4.1 Experiment 1

Considering how to carry out behaviors that do not waste time and energy in the workplace, a task robot should employ ‘quickly’ and ‘smoothly’ as behavioral criteria. Two kinds of behaviors are selected as the model’s basic behaviors in this paper: Minimum Distance Movement and Minimum Maneuvering Movement. The former is a common behavioral pattern for touching a target, where the trajectory of a fingertip is a straight line. For example, a trajectory set is \((P_1, P_2, P_3, \ldots, P_N)\), where \(P_1\) is the start and \(P_N\) is the goal.

\[
\sum_{i=1}^{N-1} d(P_i, P_{i+1}) = \text{minimum distance,}
\]

where \(d(P_a, P_m)\) is the distance from point \(P_a\) to point \(P_m\). The latter expresses minimum maneuvering in terms of turning angles. The advantage of Minimum Maneuvering Movement is that the robot can reduce the amount of energy required for moving to the goal. In this study, the trajectory of behavioral patterns is given not as a programmed one but by a genetic algorithm (GA), as explained in Subsection 2.1. Candidates for the evaluation criteria adopted in this algorithm are the following:

**Minimum Distance Movement criterion:**

\[
E_d = \int_0^t \left( (x_f(t) - x_f(t+1))^2 + (y_f(t) - y_f(t+1))^2 + (z_f(t) - z_f(t+1))^2 \right) dt
\]

**Minimum Maneuvering Movement criterion:**

\[
E_g = \int_0^t \sum_{j=0}^{N-1} \left( \dot{\theta}_j(t) - \dot{\theta}_j(t+1) \right)^2 + \left( \dot{\phi}_j(t) - \dot{\phi}_j(t+1) \right)^2 + \left( \dot{\lambda}_j(t) - \dot{\lambda}_j(t+1) \right)^2 \right) dt
\]

Here, \(t\) indicates the total time taken for moving from a start point to a goal point. \(N\) denotes the number of joints. \(\theta\), \(\phi\), and \(\lambda\) show the rotation angles of the X-Y rotation, Y-Z rotation and Z-X rotation, respectively.

Figure 7 shows the windows presented by the implemented system. The number of individuals is set to 9 at each generation. The user can select two preferred behaviors from among them as the candidates for the next generation.

![Fig. 7 Windows in the implemented system](image)

In experiment 1, through the two stages included in the design system, our ‘imitation robot’ learned the two behavioral criteria, Minimum Distance Movement and Minimum Maneuvering Movement, from the ‘model robot’. As a result, a novel behavior is designed that can be performed quickly and smoothly in the workplace as shown in Figure 8 and Figure 9. In Figure 9, (a) and (b) are the behaviors taught by the ‘model robot’, (c) is an example of simply combining the two model behaviors, and (d) shows the result designed using the system. On comparing them, we found that the behavior designed using our system is faster and smoother than either the model behaviors or the behaviors simply combined from existing behaviors. If the user does not design the behavior using the system, he or she would probably adopt a behavior similar to (c). Thus, we believe the
proposed system can create novel behaviors and can expand the range of possibilities available to the user.

![Fig. 8 A designed lifting behavior](image)

**Fig. 8 A designed lifting behavior**

**Fig. 9 Four kinds of lifting behavior**

4.2 Experiment 2

The method used in experiment 2 is similar to that used in experiment 1, except that we do not set any target for the behavior. As a result, a variety of behaviors was created.

![Fig. 10 Pitching behavior designed in this study](image)

**Fig. 10 Pitching behavior designed in this study**

As an example, Figure 10 shows the behavior of pitching a baseball. More interesting robotic behavioral designs can be created as far as the system is applied in practice.

5. Conclusion

We have described a prototype behavior-creating support system for robots that uses evolutionary techniques. New robotic behaviors are created by means of evolutionary techniques and presented to the user. As a result of the interaction between the user and the system, it becomes possible to utilize the user’s creativity and to design novel behaviors. Since the system does not simply combine existing behaviors but creates novel behavioral patterns depending on newly acquired criteria, a wide variety of robotic behaviors can be created.

A system for creating novel behaviors depending on a newly acquired criterion composed of Minimum Distance Movement and Minimum Maneuvering Movement has been implemented as an example. Through several interactive sessions, various robotic behaviors were created. In addition, the creation of dance patterns or sports patterns was also possible. It is very interesting that behaviors were created that may not have been envisaged easily by a human, such as a deformed minimum-angle-changing behavioral pattern.

We should mention here that if the designer has a concrete overall behavioral pattern already in his or her mind, it is more effective to program the behavioral pattern directly. However, if the behavioral pattern is not concrete, or if the designer is an inexperienced programmer, the described system will be more effective for creating the behavioral design.

Although there is room for improvement of the prototype, such as through acquiring new kinds of criteria, and the selection of terminals for genetic programming, we consider that the concepts proposed in this paper will lead a new method of behavioral design for robot by computers.

References


