이동 로봇 행위의 실시간 진화

이재구† · 심인보* · 윤중선**

Realtime Evolutionary Learning of Mobile Robot Behaviors

Jaegu Lee, Inbo Shim and Joongsun Yoon

Key Words : Realtime Learning(실시간 학습), Evolutionary Algorithms(진화 기법), Evolutionary Robotics(진화 로봇), Interactive Evolutionary Algorithm(인터액티브 진화 알고리즘)

Abstract

Researchers have utilized artificial evolution techniques and learning techniques for studying the interactions between learning and evolution. Adaptation in dynamic environments gains a significant advantage by combining evolution and learning. We propose an on-line, realtime evolutionary learning mechanism to determine the structure and the synaptic weights of a neural network controller for mobile robot navigations. We support our method, based on (1+1) evolutionary strategy which produces changes during the lifetime of an individual to increase the adaptability of the individual itself, with a set of experiments on evolutionary neural controller for physical robots behaviors. We investigate the effects of learning in evolutionary process by comparing the performance of the proposed realtime evolutionary learning method with that of evolutionary method only. Also, we investigate an interactive evolutionary algorithm to overcome the difficulties in evaluating complicated tasks.

1. 서 론

진화(evolution)와 학습(learning)은 시간과 공간에서 서로 다른 두 가지 형태의 생물학적 적응 방법이다. 진화는 기하학적으로 분산된 개체군을 기반으로 하는 선택적인 재생과 대체의 과정으로 볼 수 있다. 반면에 학습은 각각의 독립 개체가 자신의 삶 동안의 다른 형태로의 수정으로 볼 수 있다. 진화와 학습은 다른 시간 스케일에서 작용한다. 진화는 여러 세대에 걸친 비교적 늦은 환경의 변화에 대응할 수 있는 종의 적응인데 반하여, 학습은 그 세대에서 불확실한 환경의 변화에 잘 적응할 수 있는 개체의 수정이다[1]-[4].

† 부산대학교 지능기계공학과
* LG 전자
** 책임저자, 부산대학교 기계공학부
E-mail : jsyoon@pusan.ac.kr
TEL : (051)510-2456 FAX : (051)510-3084

진화 로봇공학(evolutionary robotics)은 사람의 간섭 없이 또는 최소한의 간섭만을 허용하며 불확실하고 동적인 환경에서 스스로 적응해 나가는 로봇 시스템의 개발을 목적으로 한다. 진화 로봇은 로봇의 요소들이 인공 염색체로 표현되는 인공 진화(artificial evolution)에 기반을 두고 있다. 진화 로봇은 여러 세대에 걸쳐 천천히 환경에 적응하기 때문에 주어진 공간에서 최대의 평가 값에 의해 진화된 개체들도 환경 변화에는 신속히 대응하지 못하는 문제점을 가진다[1][2][5][6].

이러한 문제점에 대한 대안으로, 신경 회로망(neural networks) 구조를 가지는 이동 로봇의 제어기를 결정할 (1+1) 진화 전략(evolutionary strategy)에 의한 제어기의 진화적 학습(evolutionary learning) 방법을 제안하였다[7]-[11]. 제안된 방법에서 학습과 진화는 실제 로봇을 통해 on-line 방식으로 이루어지며, 장애물 회피 문제를 통해 제안된 방법의 유용성을 검증한다. 평가 기준의 설정 문제 해결을 위하여, 수학적으로 표현되는 평가함수에 의한 평가대신 평가 과정에 인간 평가자(human evaluator)의 개입을 허용하는(heuristic) 인

터액티브 진화 알고리즘(interactive evolutionary algorithm) 방법에 대해서도 논의하고자 한다[12].

2. 진화 로봇공학

2.1 진화 로봇

진화 로봇은 로봇의 제어 구조나 전자 회로와 같은 로봇의 요소들이 인공 염색체(artificial chromosome)로 표현되는 인공 진화에 그 기반을 두고 있다. 인공 진화에 있어서 한 세대의 개체군 중에서 최상의 수행 능력을 나타내는 몇몇 개체들은 다음 세대의 개체 생성을 위해 선택되고, 선택된 개체들은 재생과 교배, 돌연변이를 통해 다음 세대의 개체들을 생성하게 된다. 그리고 이러한 진화 과정은 개체가 요구되는 특성을 나타내거나 더 이상의 진화가 이루어지지 않을 때까지 반복 수행된다[5][6].

로봇의 인공적인 진화는 여러 개체를 포함하고 있는 개체군의 반복적인 평가를 해야 하기 때문에 많은 시간을 필요로 할뿐만 아니라, 기계적인 강인성, 효율적인 결과분석, 에너지의 연속적인 공급, 그리고 적절한 평가 함수의 설계 등 몇 가지 요구되는 사항들이 있다. 이러한 요구사항을 만족하기 위해서 소형화, 모듈화, 확장성, 인터페이스 등을 기본개념으로 만들어진 로봇이 Fig. 1 의 Khepera 이다[1][5].

본 연구의 실험에 사용된 Khepera 는 직경 55mm, 무게 70g 의 소형 이동 로봇으로서, 전방에 6 개, 후방에 2 개의 적외선 센서가 부착되어 주변의 상황을 인지하고 DC 모터로 구동되는 2 개의 바퀴를 가지고 있다. 이 로봇은 Motorola MC683313 마이크로 프로세서가 탑재되어 있고 4 개의 충전 가능한 NiCd 전지가 있어 약 30-40 분 정도의 에너지 자율성을 가진다. 또한 RS232 시리얼 라인을 통해 데이터 전송뿐만 아니라, 연속적인 에너지 공급도 가능하다.

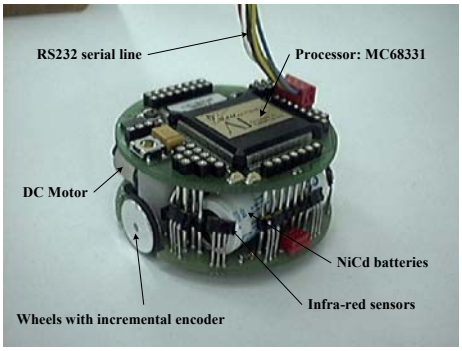

Fig. 1 The miniature mobile robot "Khepera"

2.2 제어기와 주행 환경

로봇의 제어기는 센서의 입력과 모터의 출력을 직접 연결시켜주는 신경 회로망 구조를 가지며 Fig. 2 와 같다[1][5].

이 구조는 Braitenberg's vehicle 의 3c 형태의 제어기 구조에 순환 연결(recurrent connections)과 시그모이드(sigmoid) 함수를 추가시켜 구성된다. 이러한 구조는 Braitenberg's vehicle 이 취약한 대칭인 장애물의 회피를 목적으로 고안되었다[1][5][13].

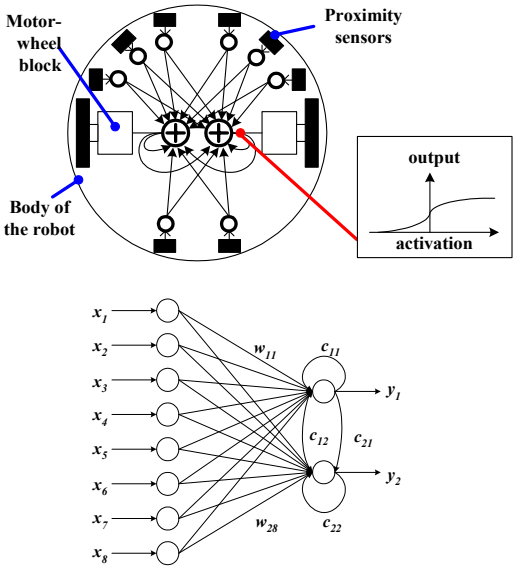

Fig. 2 A neural network controller

이 제어기의 출력 값은 식 (1)과 같이 표현된다.

$$y_i^t = F\left(\sum_i w_{ij} x_j^t + \sum_k c_{ik} y_k^{t-1}\right) \qquad (1)$$

여기 회로망의 출력 값이며, 모터의 제어입력이 된다. w_{ij} 는 입력 노드와 출력 노드를 연결시켜주는 연결강도(connection weights)이고, x_j^t 는 매 샘플링 시간마다의 입력 값이다. 그리고 c_{ik} 는 순환 연결강도이고, y_k^{t-1} 는 이전 샘플링 시간의 출력 값이다. F 는 신경 회로망의 활성화 함수(activation function)로써 양극성 시그모이드 함수를 사용하였으며, 식 (2)와 같다[1][14].

$$F(\Sigma) = \frac{1-e^{-\Sigma}}{1+e^{-\Sigma}} \qquad (2)$$

장애물을 피하여 직진 주행 실험은 Fig. 3 과 같은 환경 하에서 이루어지도록 하였다. 주행 환경은 모두 목재에 흰색의 종이를 붙인 동일한 장애물로 이루어져 있으며 여러 가지 기하학적 구조를 포함하고 있다. 주행 공간의 최대 크기는 110cm

* 110cm 이다. 로봇은 PC 와 연결되어 있어 모든 진화 작업과 신경 회로망의 연산은 PC 에서 이루어진다. 시리얼 라인을 통해 신경 회로망에 의해 나온 출력 값은 로봇으로 전달되고, 로봇으로부터의 센서 입력 값은 PC 로 전달된다.

Fig. 3 Experimental setup for robot navigation

2.3 평가 함수

로봇을 구동시키는 제어 입력값은 매 100ms 마다 출력되는 신경 회로망의 출력값이 된다. 이 구동 결과 측정되는 로봇의 양 바퀴의 구동속도나 로봇과 주변 물체와의 거리와 같은 로봇의 상태 값으로 로봇 제어의 성능을 평가할 수 있다.

장애물 회피 직진 주행을 위한 평가함수는 식 (3)과 같다[1][5].

$$\Phi = V(1-\sqrt{\Delta v})(1-i)$$
$$0 \le V \le 1$$
$$0 \le \Delta v \le 1$$
$$0 \le i \le 1 \qquad (3)$$

여기서 V 는 두 바퀴의 회전속도의 절대값의 평균이다. 활성화 함수의 범위가 [-1, 1]이기 때문에 V 는 0 과 1 사이의 값이 되며, 두 바퀴의 회전 방향에 상관 없이 회전 속도가 빠를수록 1 에 가까운 값을 갖는다. ΔV 는 두 바퀴 속도 차의 절대값이다. $(1-\sqrt{\Delta v})$는 바퀴의 회전 방향이 일치되고 값이 같을수록 1 에 가까운 값을 나타내다. 제곱근을 사용한 것은 작은 속도차도 잘 반영시키기 위한 것이다. 이 두 요소의 조합으로써 로봇은 정방향 또는 역방향으로 최대의 속도를 가지고 직진 주행을 할 때 높은 평가 값을 나타낼 수 있다.

마지막으로 (1-i)는 로봇이 가지고 있는 8 개의 적외선 센서 중에서 가장 큰 값을 가지는 센서의 입력 값이다. 로봇이 가지는 적외선 센서는 5cm 의 검출범위를 가지며, 물체가 가까울수록 1 에 가까운 값을 가진다. 그러므로 주변에 장애물이 멀리 떨어져 있을 때, 1 에 가까운 값을 가진다.

이러한 세가지 요소들은 로봇의 빠른 움직임, 직진 주행, 장애물 회피의 행위들을 촉진시키지만 방향성에 영향을 주지는 못한다.

3. 로봇 행위의 진화

이동 로봇이 장애물을 피하여 직진 주행을 하는 행위를 진화하고자 한다. 신경 회로망의 구조를 가지는 행위 제어기의 연결강도를 진화의 대상으로 한다.

3.1 행위 진화 알고리즘

로봇을 구동시키기 위한 제어기는 Fig. 2 와 같은 센서와 모터를 바로 연결하는 신경 회로망 구조를 가진다. 제어기의 연결강도를 진화시키기 위하여 사용된 진화 알고리즘(evolutionary algorithm)에 의한 제어시스템은 Fig. 4 와 같다.

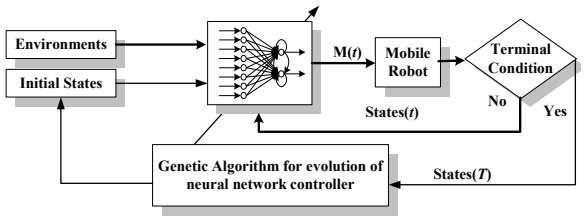

Fig. 4 Control system for evolutionary robot

제어기의 연결강도는 Fig. 5 와 같은 인공 염색체로 표현된다. 평가는 매 샘플링 시간 100ms 마다 하되, 한 개체의 평가 값은 100 회를 반복한 후의 평균값으로 한다. 제어시스템에 사용된 유전자 알고리즘(genetic algorithm)의 설정 및 파라미터 값들은 Table 1 과 같다.

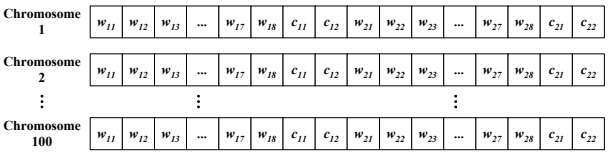

Fig. 5 Artificial chromosomes for controller connection weights

Table 1 Parameters for genetic algorithm

Parameters	**Values**
Population size, N	100
Chromosome length, *l*	20
Initial weight range	-0.5 ~ +0.5
Life length	100 actions
Action duration	100ms
Selection	Ranking + elite
Crossover	Multipoint Crossover
Mutation	N(0,1)
Crossover Rate, Pc	0.1
Mutation Rate, Pm	0.2

개체군(population)의 크기는 100 이며, 염색체의 길이는 20 이고, 초기 weight 값의 범위는 -0.5 ~ +0.5 이다. 각 개체의 센서 입력 값과 네트워크 활성화에 의한 출력 값은 매 100ms 마다 갱신되고, weight 의 변화 없이 총 10 초 동안의 평가를 받는다.

한 세대가 끝난 이후 다음 세대의 개체 생성을 위해 평가가 끝난 개체를 평가 값의 크기에 따라 순서를 매긴 후, 순서에 따라 개체군의 크기의 반만큼의 개체, 즉 N/2 개의 개체를 선택한다. 선택된 개체 중 최고의 평가 값을 가지는 개체와 임의의 개체를 선정하여 교배확률 Pc 를 적용시켜 교배를 시키고, 그 중 하나의 개체만을 택한다. 이를 돌연변이 확률 Pm 을 통해 돌연변이 개체를 만들어 내는데, 돌연변이는 평균이 0 이고, 표준편차가 1 인 표준분배 N(0,1)에 의해서 이루어 진다. 이러한 과정은 새롭게 생성되는 개체군의 크기가 초기 개체군의 크기와 같을 때까지 반복된 후, 다음 세대의 개체들로서 역할을 하게 된다. 진화과정은 100 세대에 걸쳐 반복된 후 종결된다.

3.2 행위의 진화

장애물을 회피하여 빠른 속도로 직진 주행을 할 수 있는 이동 로봇의 제어기를 진화시키는 실험을 수행하였다. 진화는 여러 세대에 걸친 비교적 늦은 환경의 변화에 대응할 수 있는 적응의 한 형태이기 때문에 100 세대 동안 진화 과정을 반복 수행하였다. 한 개체의 평가 시간은 10sec 이고, 한 세대는 1000sec, 전체 실험 시간은 약 27.8 hours 이 걸렸다.

실험 결과 행위 진화 성능 그래프는 Fig. 6 과 같으며, 각 세대에 있어 모든 개체의 평균 평가 값과 최고 성능을 나타낸 개체의 평가 값을 나타내고 있다.

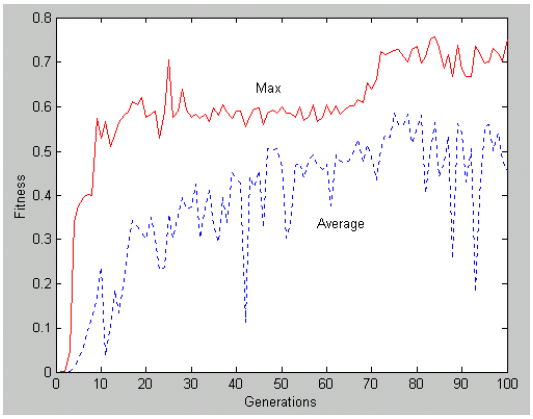

Fig. 6 Average fitness of the population and fitness of the best individual

최상의 개체의 경우 세대가 증가함에 따라 평가 값이 증가하는 경향을 보이며, 평가 값 또한 높은 값을 나타냈다. 이는 실험 환경이 비교적 열린 공간이 많기 때문이긴 하지만 개체들의 평균값들도 증가하는 경향을 볼 수 있다. 그러나 세대에 따른 개체들의 평균 평가 값들이 심한 변동을 보이는 것을 확인할 수 있었는데, 이는 그 세대에서 개체들이 대칭형 장애물 공간에서 잘 벗어나지 못했기 때문이다. 진화는 여러 세대에 걸쳐 천천히 환경에 적응하기 때문에 열린 공간에서 최대의 평가 값을 가졌던 개체들로부터 진화되어온 다음 세대의 개체들은 변화한 환경에 제대로 대응하지 못하고 낮은 평가 값을 나타낸다는 문제점을 가진다.

4. 로봇 행위의 진화적 학습

이동 로봇의 장애물 회피 직진 주행하는 제어기의 진화의 경우 세대당 1000 초씩 100 세대 총 30 여시간이 걸렸고 Mondada 와 Floreano 의 공을 잡는 제어기의 진화의 경우 세대당 39 분씩 100 세대 총 65 여시간이 걸렸다[15]. 진화 연산의 효율성(efficiency)과 구현성(feasibility)이라는 상충의 문제는 기존의 진화 로봇공학의 연구에 가장 큰 걸림돌이 되고 있다. 이에 대한 대안으로서 실시간 진화적 학습 방법을 제안한다[7][9][10][11].

4.1 행위의 진화적 학습

본 연구에서 '학습'은 개체 스스로가 자신의 적응성을 높이기 위해 신경회로망으로 구성된 제어기의 가중치 값을 lifetime 동안 수정하는 것을 지칭하기 위해 사용된다. 즉 개체는 변화하는 환경에 잘 적응하기 위해서 매 샘플링 시간마다 신경회로망의 가중치를 조정하는 학습 과정을 수행하게 된다[6][8].

학습을 위한 방법으로는 (1+1) 진화 전략(evolutionary strategy)을 사용하였다[14]. 진화 전략에 있어 한 세대는 제안된 제어 알고리즘에서의 샘플링 시간이 되고, 매 샘플링 시간마다 돌연변이 연산에 의해 신경회로망의 가중치는 주어진 환경에 잘 적응할 수 있도록 조정된다. 진화적 학습을 위한 제어 시스템은 Fig. 7 과 같다[11].

진화 전략에서 개체는 $(x,\ \sigma)$와 같이 한 쌍의 실수 벡터로 표현된다. x 는 탐색공간 안의 위치 벡터이고 σ 는 표준편차 벡터이다. 본 연구에서는 x 는 신경회로망 제어기의 연결 가 중치 값들이 되며 세대 t 에서의 개체는 식 (4)와 같이 표현된다.

$$\left(X_i^t, \sigma_i^t\right) \quad \forall i \in \{1, K, n\} \tag{4}$$

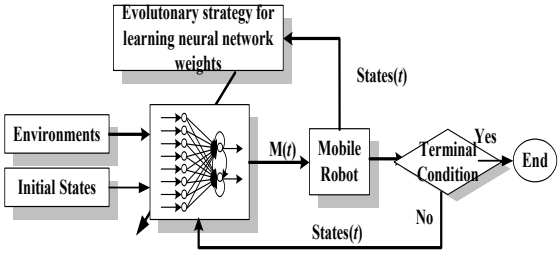

Fig. 7 Control system of evolutionary robot with realtime learning

여기서 n 은 가중치의 개수이며, 본 연구에서는 20 이 사용되었다. 진화 알고리즘은 지역적 탐색과 전역적 탐색과정을 통해 탐색대상의 해답을 찾는다. 유전자 알고리즘(genetic algorithm)은 교배(crossover)와 돌연변이(mutation) 연산을 통해 이를 가능하게 한다. 이에 반해 진화 전략(evolutionary strategy)은 돌연변이 연산의 표준 편차 값의 크기에 따라 전역 탐색과 지역 탐색이 가능해 진다[14].

학습을 위한 제어기의 구조는 Fig. 2 와 동일하다. 학습 대상은 외부로부터 자극을 받는 노드의 연결 가중치이다. 외부의 장애물을 감지한 센서와 연결된 가중치들과 순환 연결구조의 가중치들이 학습대상이 된다[11].

신경 회로망의 가중치 조정을 위하여 돌연변이 연산에 의한 자식 개체(offspring)는

$$x^{t}_{offspring} = x^{t}_{parent} + N(0,\sigma) \tag{4}$$

와 같이 조정되고 표준 편차는

$$\sigma^{t}_{offspring} = \sigma^{t}_{parent} \cdot e^{N(0,\Delta\sigma)} \tag{5}$$

와 같이 조정된다. σ 의 변화에 따라 탐색 영역은 Fig. 8 과 같이 달라질 수 있으며, 평가 함수에 의한 평가 값이 이전 세대보다 높으면 σ 를 감소시키고 반대로 평가 값이 낮으면 σ 를 증가시킨다. σ 의 증감은 식 (5)의 지수함수에 의해 결정된다. σ 를 평가 값의 변화에 따라 변화시키기 위해서 식 (5)를

$$\begin{aligned} &if \;\; fitness^{t-1}_{offsping} \geq fitness^{t}_{offspring} \\ &then \;\; \sigma^{t}_{offspring} = \sigma^{t}_{parent} \cdot e^{abs(N(0,\Delta\sigma))} \\ &else \;\; \sigma^{t}_{offspring} = \sigma^{t}_{parent} \cdot e^{-abs(N(0,\Delta\sigma))} \end{aligned} \tag{6}$$

처럼 수정하여 사용한다. Δσ 는 σ 의 변화량을 결정하는데 본 실험에서는 0.3 으로 난수는 ±1 사이에서 발생한다. 평가 함수는 식 (3)과 같다.

식 (7)과 같이 새롭게 생성된 자식 개체의 평가 값이 현재의 부모 개체의 평가 값보다 높으면, 현재의 자식 개체를 다음 세대의 부모 개체로 선택하고, 반대로 부모 개체의 평가 값이 높으면 현재의 부모 개체가 다음 세대의 부모 개체가 된다.

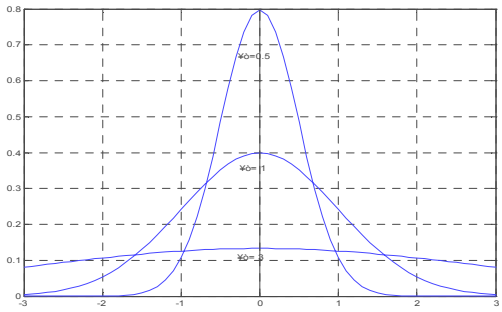

Fig. 8 Variation of search space due to σ changes

$$\begin{aligned} &if \quad fitness^{t}_{parent} < fitness^{t}_{offspring} \\ &then \quad (X^{t+1}_{parent}, \sigma^{t+1}_{parent}) = (X^{t}_{offspring}, \sigma^{t}_{offspring}) \\ &else \quad (X^{t+1}_{parent}, \sigma^{t+1}_{parent}) = (X^{t}_{parent}, \sigma^{t}_{parent}) \end{aligned} \tag{7}$$

4.2 행위 진화

신경회로망의 가중치는 매 100ms 의 샘플링 시간마다 조정되고, 10,000 번을 반복 수행한다. 평가 값이 0.5 이하이면 가중치 조정 학습이 수행되고 0.5 이상이면 가중치 조정없이 작동하게 된다.

Fig. 9 는 매 샘플링 시간마다의 평가 값, 즉 진화 전략에 있어 각 세대마다의 평가 값을 나타내고 있다. Fig. 9 에서 개체의 전체적인 평가 값의 평균은 종결 조건인 0.5 를 유지하고 있다. 학습 단계 0~300 까지는 빠른 속도로 직진하는 요소의 증가로 전체적인 평가 값이 증가하였다. 이후로는 장애물 회피 능력이 증가하여 전체적인 평가 값이 종결 조건인 0.5 에 가까운 값을 나타내었다.

하지만 종결조건이 만족되어 학습이 종결된 후에는 어느 정도의 평가 값의 증가가 보이지만 장애물에 접근함에 따라 다시 평가 값이 낮아졌다. 다시 말해서 학습을 통해 개선된 개체는 현재 주어진 상황에 대해서 최적화되기 때문에 환경이 바뀌고 로봇의 상태 값이 바뀌게 되면 이전의 평가 값을 유지할 수 없게 된다. 그러나 학습을 통해 곧바로 변화된 조건에 맞는 가중치를 찾아내게 되어 평가 값을 상승시킬 수 있다.

주변 환경 변화에 대한 적응성을 알아보기 위해 3500 번째부터는 열린 공간을 줄이고 장애물을 더 첨가 하였다. 그 결과로 몇 단계에 걸친 평균적인 평가 값이 낮아지긴 하였지만 학습을 통해 이를 극복하고 다시 평가 값을 증가시키는 것을 볼 수 있었다.

실험에서 학습은 변하는 환경에 대한 높은 적응성을 나타내지만, 학습된 행위는 부분적으로 최적화된 것이기 때문에 원하는 행위의 달성에 있어 불확실성을 가지는 문제점이 있다.

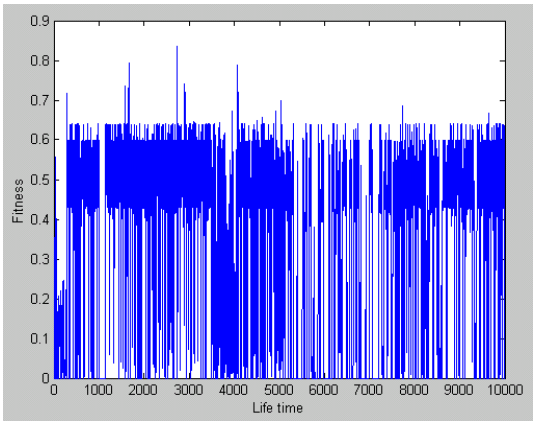

Fig. 9 Fitness of an individual vs. action duration

4.3 인터액티브 진화 알고리즘

진화와 학습의 상호작용을 통해 각각의 문제점을 개선하기 위한 방법으로 Fig. 10 과 같은 통합 제어 시스템을 제안한다. 이 시스템에는 진화 과정 중에 개체의 학습 과정을 평가할 수 있는 방법으로 인터액티브 진화 알고리즘과 적응성을 높이기 위한 네트워크의 자기 구조화(self-organization) 방법이 적용된다[11][12][16][17].

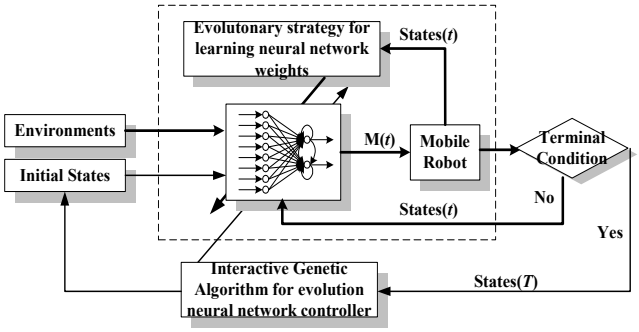

Fig. 10 The integrated control system for robot's evolution and learning

5. 결 론

개체는 진화와 개체 학습을 통하여 부모 세대로부터 독특한 특성을 물려 받는다. 학습을 통해 주어진 환경에 잘 적응한 개체들은 다음 세대에 선택될 확률이 높으며, 학습은 진화를 가속시킬 수 있다.

진화 알고리즘에 의한 이동 로봇 제어기의 진화와 진화적 학습 방법에 대해 논의 하였다. 진화적 학습을 통한 제어기 설계 방법은 동적으로 변화하는 환경에 대한 뛰어난 적응성을 나타낸다. 하지만 진화를 통한 방법에 비해 안정적이지 못하다는 결과를 얻을 수 있었다. 학습에 의한 적응성과 그것을 평가하여 진화함으로써 안정성을 획득할 수 있는 통합적인 진화-학습 시스템이 요구된다.

참고문헌

(1) Nolfi, S. and Floreano, D., 2000, *Evolutionary Robotics*, Cambridge: The MIT Press.
(2) Nolfi, S. and Floreano, D. 1999, "Learning and Evolution," *Autonomous Robots*, Vol. 7, No. 1, pp. 89-113.
(3) Zimmer, C., 2001, *Evolution: The Training of the Idea*, New York: Harper Collins Publishers.
(4) Bentley, P. J., 2001, *Digital Biology: How Nature is Transforming Our Technology and Our Lives*, New York: Simon & Schuster.
(5) Mondada, F. and Floreano, D., 1999, "Evolution and Mobile Autonomous Robotics," *Towards Evolvable Hardware: The Evolutionary Engineering Appoach*, Berlin: Springer-Verlag, pp. 221-249.
(6) Floreano, D. and Urzelai, J., 2000, "Evolutionary Robots with On-line Self-organization and Behavioral Fitness," *Neural Networks*, Vol. 13, pp. 431-443.
(7) Saravana, N. and Fogel, D.B., June 1995, "Evolving Neural Control System," *IEEE Expert*, Vol. 10, No. 3, pp. 23-37.
(8) Urzelai, J. and Floreano, D., 2000, "Evolutionary Robots with Fast Adaptive Behavior in New Enviroments," *Proceedings of the Third International Conference on Evolvable Systems: From Biology to Hardware*, Berlin: Springer-Verlag, pp. 241-251.
(9) Son, H. and Yoon, J., 2000, "Evolving Neural Network for Realtime Learning Control," *Proceedings of 15th Korea Automatic Control Conference*, Hyundai Learning Center, Yongin.
(10) Sohn, H., 2002, *Study of a Real-time Learning*, Master thesis, Pusan National University.
(11) Shim, I. and Yoon, J., 2002, "Evolutionary Learning of Mobile Robot Behaviors," *Proceedings of KFIS 2002 Fall Conference*, Pukyoung National University: pp. 207-210.
(12) Yoon, J. 2000, "Evolutionay Robotics based on Interactive Technology," *Proceedings of 15th Korea Automatic Control Conference*, Hyundai Learning Center , Yongin.
(13) Braitenberg, V., 1984, *Vehicles: Experiments in Synthetic Psychology*, Cambridge: The MIT Press.
(14) Negnevitsky, M., 2002, *Artificial Intelligence*, San Francisco: Addison-Wesley.
(15) Jordan, P., Hod, L., Pablo, F., Sevan, F. and Greg, H.,1999, "Coevolutionary Robotics," *Proceedings of the First NASA/DoD Workshop on Evolvable Hardware*, IEEE Computer Society, Los Alamitos, CA: pp. 208-216, 1999.
(16) Patel, M. J., Honavar, V. and Balakrishnan, K., 2001, *Advances in the Evoltionay Synthesis of Intelligent Agent*, Cambrige: The MIT Press.
(17) Son, H., Kang, J. and Yoon, J., 1999, "Interactive Composition of a Genetically Evolved Fractal Music," *Proceedings of the 1999 International Conference on Mechatronic Technology*, Pusan National University: pp. 422-427.