행위 기반 로봇에서의 행위의 자동 설계 기법

A Self-Designing Method of Behaviors in Behavior-Based Robotics

윤 도 영, 오 상 록, 박 귀 태
(Do-Young Yoon, Sang-Rok Oh, Gwi-Tae Park)

Abstract : An automatic design method of behaviors in behavior-based robotics is proposed. With this method, a robot can design its behaviors by itself without aids of human designer. Automating design procedure of behaviors can make the human designer free from somewhat tedious endeavor that requires to predict all possible situations in which the robot will work and to design a suitable behavior for each situation. A simple reinforcement learning strategy is the main frame of this method and the key parameter of the learning process is significant change of reward value. A successful application to mobile robot navigation is reported too.

Keywords : behavior-based robot, design automation, reinforcement learning, robotic behavior

I. 서론

행위 기반 로봇(behavior-based robot)은 로봇이 실행하게 될 모든 동작을 일정한 종류와 수의 행위(behavior)들로 나누어 설계하여 두었다가 각각의 상황에 직면하게 되면 중앙에 사람의 뇌와 같은 주변 상황의 모델링(modeling)을 통한 인지(cognition)와 추론(reasoning)의 과정 없이 곧바로 설계된 행위 들을 적절히 선택·조합하여 로봇의 동작을 결정하는 반사적인 응답(reactive response)방식으로 동작한다. 이 방식의 핵심 사항은 두 가지로 구분할 수 있다. 하나는 로봇이 처하게 되는 상황들을 미리 예측하여 필요한 행위 들을 결정하고, 그 각각의 행위를 구현하는 행위 분해(behavior decomposition)과정과, 설계된 행위들을 상황에 맞게 적절히 선택하고, 조합하여 로봇의 최종 동작을 결정하는 과정인 행위 조정(behavior coordination) 과정이다. 물론 각각의 행위가 어떠한 형태로 설계되느냐에 따라서 행위의 조정 방법이 달라지므로 이 두 과정은 밀접하게 연관되어 있는 작업이다. 행위기반 로봇은 인지(cognition) 과정이 없이 센서의 입력에 대한 지각(perception)에 대하여 곧바로 반사적으로 반응하므로 각각의 행위는 센서와 동작(action)이 밀접하게 연결된 지각 모터(sensory motor)의 형식을 띄는 것이 일반적이다. 행위의 분해는 설계자마다 다른 방식을 취한다. 기존의 대표적인 행위의 분해의 예를 보면 행위를 계층으로 분류하여 하위 계층의 간단한 행동에서부터 상위 계층의 보다 복잡하고 세련 된 행위로 설계자가 직접 만들어 나가는 형식[2], 동일한 중요성을 가지는 수평적인 기본적인 (primitive) 행위들로 분해한 뒤 이들의 벡터 합으로 새로운 행위들을 만들어 나가는 방식[3], 자유로운 형식으로 행위를 설계자가 분해한 뒤 행위의 각각에 조정(coordination)을 위한 파라미터를 주고 이를 상황에 맞게 변화시키는 방식[4], 행위의 분해 시 조정에 관련된 파라미터의 값을 multi-value로 주어 조정 시 퍼지 추론이 가능하게 하는 방식[5] 등을 들 수 있다. 이 들 방식에서는 모두 로봇이 동작하면서 일어날 수 있는 모든 상황을 미리 예측하여 각각의 상황에 맞는 고유한 종류와 수의 행위들을 사전에 설계하게 된다. 이 과정은 설계자가 사전에 많은 시간을 들여 행위를 설계해야 하며 로봇이 처한 상황이 미리 예측된 상황이 아니면 적절한 행위를 취할 수 없다. 본 논문에서는 이러한 과정을 자동화하고, 보다 효율적인 행위들로 분해하는 방법을 제안하기로 한다. 이를 위해 우선 생각해야 할 것은 행위를 나누고 결정하는 기준이 무엇인가를 명확히 하는 것이다. 그 기준은 바로 상황(case 또는 situation)이다. 기존의 방식은 설계자들이 직접 로봇이 동작하면서 직면하게 될 상황들을 미리 예측하여 나열하였다. 그리고 그 상황들에 대처할 수 있는 행위들을 생각하고, 그 것들을 구현하였다. 그러나 여기서 말하는 상황이란 것을 좀더 생각해 볼 필요가 있다. 로봇에게 있어서 중요한 것은 실제 존재하게 되는 상황 그 자체가 아니라 로봇이 느끼는 센서를 통한 이미지이다. 또한 상황을 나눌 때 중요한 것은 설계자가 보는 것이 아니라 로봇이 다른 상황이라고 지각(perception)을 하는 것이다. 실제 상황 그 자체나, 또는 설계자인 사람이 판단하는 상황들 중에 어떤 것들은 로봇이 센서를 통하여 느끼지 못하거나, 판별하지 못할 수 도 있다. 이러한 이유에서 본 연구에서는 상황의 판단은 로봇 스스로가 해야 하고, 그에 맞는 동작 방식 역시 스스로 구현하여야 한다는 관점에서 새로운 방법을 제시하였다. 설계자의 입장에서는 행위를 자동으로 설계함으로써 로봇이 동작하면서 일어날 수 있는 모든 상황을 미리 예측하여 각각의 상황에 맞는 행위를 직접 설계해야 하는 다소 복잡하고 세심한 작업의 부담을 덜 수 있고, 로봇의 입장에서 설계가 이루어지므로 사전에 실제 로봇이 느끼지 못하는 상황이나, 불필요한 행위를 설계하는 오류를 막을 수 있다. 이렇게 함으로써 설계 자동화와 로봇에게 의미 있는 행위의 구현이 가능하다. 본 논문에서는 일반적인 종류의 임무를 수행하는 로봇에 적용할 수 있는 행위의 범용적인 설계 방법을 제시하고, 2차원 이동 로봇(mobile robot)의 자율주행에 적용하여 본 결과를 통해 제안한 설계 방법의 유효성을 보였다.

논문접수 : 2001. 12. 10., 채택확정 : 2002. 4. 10.
윤도영 : 고려대학교-한국과학기술연구원(doyoung@kist.re.kr)
오상록 : 한국과학기술연구원(sroh@amadeus.kist.re.kr)
박귀태 : 고려대학교(gtpark@korea.ac.kr)

II. 행위의 자동 설계 방법

행위 기반 로봇에서 '행위'라고 하는 의미 있는 단위는 자극/응답의 쌍(stimulus/response pair)을 말하는데 이를 구현하는 형태는 우리가 흔히 생각하는 일반적인 의미에서의 고차원적인 행위-설계자인 인간의 추상적인 분류 능력이 강하게 가미된, 예를 들자면 '장애물 회피'라는 행위를 설계한다면 '장애물'과 '회피'에는 이미 인간의 추상적인 개념이 반영된 것이며 로봇은 이를 직접적으로 이해하지 못한다.-를 설계하는 것과 또는 상황-동작의 쌍(situation-action pair)-예를 들자면 '전방에 물체 감지' 시에는 '뒤로 후진하라'-을 찾아내어 설계하는 두 가지가 가능하다[1]. 후자의 경우가 로봇이 센서를 통하여 느끼는 상황을 기준으로 하므로 보다 로봇의 입장에 가까운 형식이다. 본 논문에서는 로봇 스스로의 입장을 채택하므로 후자의 형식으로 '행위'를 구현하였다. 본 연구에서 제시한 방식에서는, 생물체에서 볼 수 있는 최소한의 타고난 '본능'에 해당하는 행동 양식-이를 테면 사물에 부딪히는 것을 싫은 것으로 알고, 목표점 등의 우호적 사물에는 가까이 갈 수록 좋다는 것을 아는 것 등을 예로 들 수 있다.-과 상황의 인식과 상관없이 로봇의 물리적인 시스템에 의해서 주어진 가능한 동작(action)들의 집합만을 준 상태에서, 로봇 스스로 상황을 구분하고 그에 대처하는 행위를 학습을 통하여 만들어 나간다. 즉, 로봇에게 구체적인 상황이 주어졌을 때 그 상황에 맞는 동작(action)을 스스로 찾아내어 대응(mapping) 시켜나간다. 이 과정에서 가장 중요한 사항은 상황(situation)을 구분하는 것으로 그 기준이 되는 파라미터는 특정 동작을 했을 때 환경으로부터 받는 보답(reward)이다. 즉, 보답의 갑작스러운 변화를 로봇은 새로운 상황-동작 쌍-즉, 행위-의 생성 요인으로 판단한다. 이를 기준으로 다음과 같이 강화학습[6][7]을 통하여 행위를 구현한다.

1. 학습을 통한 각각의 행위의 설계

로봇은 동작을 하면서 매 동작 단계마다 자신의 센서로부터 상황을 인식한다(perception). 여기서 말하는 '상황'은 실제 환경을 말하는 것이 아니고, 각 환경에서 로봇이 받는 고유한 센서의 패턴 벡터(sensor pattern vector) 들로 이루어지는 집합이다. 학습의 수행 전에 로봇에게 주어진 동작과 실험 조건을 다음과 같이 가정한다.

조건 1 : 로봇에게는 구현된 구조에 의해서 물리적으로 정해진 가능한 동작의 집합 $A=\{a_i\}$가 주어져 있는 상태이다. 이 집합 A에 속한 원소 a_i들은 로봇 자체의 단순한 동작을 나타내는 여러 개의 파라미터로 구성된 동작벡터로서 환경과의 관계가 고려되지 않은 것들이다.

조건 2 : 주어진 상황에서 가장 적합한 동작을 찾는 것이 학습으로서, 학습이 진행되는 동안에 로봇이 A에 속하는 동작들을 수행한 후 각각의 동작 결과에 따른 보답이 원하는 수준의 값을 넘는 동작을 그 상황에서의 학습된 행위로 간주한다. 이를 위해서는 로봇이 한 동작을 수행한 후 다시 원래의 상태로 돌아와 새로운 동작을 수행해야 하는데 이를 위한 원상태로 돌아오는 메카니즘은 주어진 것으로 가정한다.

또한 같은 동작을 취하게 되는 여러 개의 센서 패턴 벡터들이 집합을 이루어 하나의 상황 S_i로 통합되어 분류가 된다. 따라서 하나의 행위는 $b_i=(S_i, a_i)$와 같은 상황-동작의 쌍으로 표현되고 이러한 형태로 설계되어 진 행위들의 집합을 B라 한다. 또한 로봇이 동작을 취한 후 그 동작에 대하여 받는 보답 $r_a(s)$는 센서 값의 함수로서 생물체의 경우 보통 본능에 의해 주어지는 것으로 실제로는 여러 형태로 줄 수 있다.

1.1. 초기화

초기 상태에서 주어진 센서 패턴 벡터 s_0에 대하여 전체 동작의 집합 A에 속한 동작 a_i들을 시도해 본다. 이 때 미리 정해진 보답(reward)의 크기 r_{th}이상의 보답을 가지는 동작을 찾는데 그 방법은 동작 벡터를 구성하는 파라미터의 변화에 따른 보답의 증가가 최대인 방향으로 파라미터를 변화시켜 가는 것이다. 즉, 그레디언트 $\partial\gamma/\partial a$의 방향으로 동작 벡터의 파라미터를 변화 시켜 나간다. 이렇게 하여 찾아진 동작 벡터를 이 첫번째 센서 패턴에서의 동작방식으로 결정하여 하나의 행위에 해당하는 첫번째의 상황-동작 쌍인 $b_0=(s_0, a_0)$를 생성하여 저장한다. 이 때 생성된 행위들을 저장하는 집합 B도 생성된다. 이 과정을 다음 (1)과 같이 정리할 수 있다.

$$\begin{aligned} &b_0=(S_0, a_0)\\ &S_0=\{s_0\}\\ &a_0=a_j \; s.t. \; r_{aj}(s_0)\geq r_{th}, \; a_j\in A\\ &B=\{b_0\} \end{aligned} \tag{1}$$

1.2 상황의 분류 및 행위의 설계

로봇이 동작을 계속해 나감에 따라 다음의 세가지 경우로 나누어 행위를 설계한다.

1.2.1 기존에 학습한 센서 패턴인 경우

새로이 입력된 센서 패턴 s_{new}가 이전의 단계에서 학습하여 저장해 놓은 행위의 상황 집합 S_i의 패턴들과 비교하여 일치하는 것이 있으면 같은 상황으로 구분하고 그 상황에서 이미 학습된 해당 동작방식을 취한다. 이 때 새로운 행위는 생성되지 않는다. 즉,

$$\text{if } \exists i \; _{s.t.} \; S_{new}\in S_i, \; (S_i, \; a_i)=b_i\in B \tag{2}$$
$$\Rightarrow\begin{cases} a_{new}=a_i \\ b_{new}=b_i \end{cases}$$

과 같이 동작한다.

새로이 입력된 센서 패턴이 기존의 센서 패턴과 다른 경우에는 그 상황에서 다시 학습을 통해 적절한 행위를 결정해야 하는데 이 경우는 다음의 두 가지로 나누어 생각할 수 있다.

1.2.2 보답의 변화가 작은 경우

센서 패턴이 새로운 패턴일 경우, 즉 $s_{new}\notin S_i, \; \forall i$, $(S_i, \; a_i)=b_i\in B$인 경우에 우선 로봇은 기존에 학습된 행위에 속하는 동작을 취해 보고 그 결과로 받는 보답이 기존에 학습된 행위를 취했을 때 받는 보답에 비해 변화가 작으면 즉, 미리 주어진 보답의 변화에 대한 기준 값을 r_{ch}라 할 때,

$r_{ai}(s_i)-r_{ai}(s_{new})\le r_{ch}$를 만족하는 행위가 하나라도 B에 존재하면 그 동작을 새로운 패턴에 대한 적절한 동작으로 판단한다. 그러나 이 경우 새로운 센서 패턴에 대해서도 이미 학습된 동작을 취하게 되므로 이를 새로운 행위로 분류하는 대신 새로운 센서 패턴을 그 동작을 취하도록 이미 학습된 센서 패턴의 집합 즉, 그 상황에 통합시킨다. 즉, (3)의 과정을 통하여 세밀하게 변화하는 여러 개의 센서 패턴 벡터들이 하나의 단일한 동작을 수행하는 의미 있는 상황으로 통합된다.

$$\begin{aligned}&\text{for } s_{new}\notin S_i,\ \forall i,\ (S_i,\ a_i)=b_i\in B,\\ &\text{if } \exists i_{s.t.}\ r_{a_i}(S_i)-r_{a_i}(s_{new})\le r_{ch},\ (S_i,\ a_i)=b_i\in B\\ &\Rightarrow\begin{cases} a_{new}=a_i\\ b_{new}=b_i\\ S_{i,new}=Si\cup\{s_{new}\}\end{cases}\end{aligned}\tag{3}$$

이렇게 새로이 겪게 되는 센서 정보에 대해서도 새로운 동작을 찾기 보다는 이전의 행위가 담당할 수 있는 상황에 속할 수 있는가의 여부를 미리 조사함으로써 불필요한 행위의 생성을 방지할 수 있다. 실제 상황에서는 대부분의 유사한 센서 패턴에 대하여 같은 종류의 동작을 취하는 경우가 많기 때문에 이 방법을 통하여 로봇은 수 많은 센서 패턴을 제한된 몇 개의 상황으로 분류할 수 가 있다.

이전의 행위 중에서 해답을 찾는 방법은 두 가지를 사용할 수 있다. 하나는 이전의 행위의 동작을 차례로 적용하다가 해답이 나오는 첫번째 행위에 편입시키는 방법과 또 하나는 학습된 모든 패턴에 대한 동작을 적용하여 최대 보답 값을 가지는 행위를 선택하는 방법이다. 전자의 방법은 가장 유사한 패턴의 동작부터 차례로 적용하여 해답이 나오면 중단하는 방법으로 속도는 빠르나 최적성(optimality)이 고려되지 않은 방법이다. 후자는 이전의 모든 패턴을 적용하므로 속도가 느려지나, 최대의 보답을 찾으므로 학습 시에 최적성이 고려되는 방법이다. 본 논문에서는 빠른 학습을 위한 첫 번째 방법을 선택하였다.

1.2.3 보답의 변화가 큰 경우

기존의 어떠한 행위로도 해답을 얻을 수 없는 전혀 새로운 센서 패턴의 경우는 로봇이 받는 보답의 변화가 r_{ch}보다 큰 경우, 즉 $r_{a_i}(S_i)-r_{a_i}(s_{new})>r_{ch},\ \forall i,\ (S_i,\ a_i)=b_i\in B$의 경우로서 새로이 행위를 설계해야 한다. 이 때는 위의 1.1의 학습방법을 그대로 사용하되, 다만 이전에 학습된 파라미터들로 구성되는 동작들을 제외한 것들에 대해서만 학습을 진행하게 된다. 즉, (4) 와 같이 새로운 행위가 설계되고 따라서 설계된 행위들의 집합 B도 새롭게 생성된 행위를 포함하기 위해 갱신된다.

$$\begin{aligned}&\text{for } s_{new}\notin S_i,\ \forall i,(S_i,\ a_i)=b_i\in B,\\ &\text{if } r_{a_i}(S_i)-r_{a_i}(s_{new})>\forall i,\ (S_i,\ a_i)=b_i\in B\\ &\begin{cases} b_{new}=(S_{new},\ a_{new})\\ S_{new}=\{s_{new}\}\\ a_{new}=a_j\ s.t.\ r_{a_i}(s_{new})\ge r_{th.}\\ \qquad a_j\in A,\ (S_j,\ a_j)=b_j\notin B\\ B_{new}=B\cup\{b_{new}\}\end{cases}\end{aligned}\tag{4}$$

1.3 학습의 반복 및 종료

위의 1.2의 과정을 반복하여 행위들을 설계하고, 더 이상의 행위의 개수의 증가가 없을 경우 즉, 행위 집합 B의 원소의 개수(cardinality) $|B|$ 가 포화될 때 학습을 종료한다.

1.4 행위의 코딩화(encoding)

학습이 끝나면 상황-동작의 쌍 $b_i=(S_i,\ a_i)$의 형태로 나열되어 저장되어 있는 행위들을 보다 통합적인 형식인 $\beta(S)\mapsto A$의 형태로 인코딩(encoding)하여 구현한다. 여기서 센서 패턴 전체의 집합을 S, 동작 전체의 집합을 A라고 할 때, β는 S에서 A로의 행위 대응함수(behavioral mapping function)가 된다[1]. 이 함수는 일반적으로 간결한(compact) 함수의 형식을 띄기 어려운 임의의 대응 형식을 가지기 쉬우며 따라서 신경망(Neural Network), If-Then rule, 2진 논리연산(boolean logic expression) 같은 도구를 이용하여 구현할 수 있다. 그림 1에 위에서 기술한 일련의 행위의 자동 설계 과정을 요약하여 나타내었다.

2. 성능 분석

2.1 행위의 조정

위의 과정을 통해서 설계된 행위들은 상황-동작의 쌍으로 구현되었으므로 별도의 조정 방법 없이 상황에 따라 곧바로 그에 대응되는 행위가 로봇의 동작을 결정한다. 이것은 결국 경쟁적 행위 조정(competitive coordination)[2] 방법의 한 형태이다. 한 상황에 처했을 때, 학습된 여러 행위 중에서 가장 적절한 하나만이 선택되어 로봇의 전체적인 동작을 유일하게 결정하기 때문이다. 물론 여기에 센서 패턴의 유사성을 판별하는 루틴 등의 조정을 위한 별도의 장치를 첨가하여 행위의 조정 성능을 높일 수도 있다.

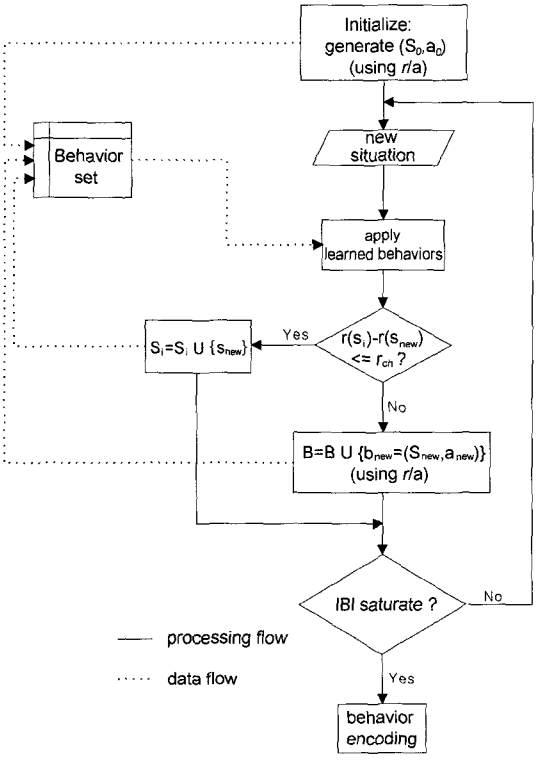

그림 1. 행위의 자동 설계 과정.

Fig. 1. Flow chart of automatic behavior design.

2.2 실효성 (feasibility of the method)

행위들의 설계 과정에서 센서 패턴의 전체 공간을 학습한 것이 아닐 수 있다. 즉, 위의 학습 과정에서 얻은 행위의 개수는 $|B|$ 가 되고, (5)에서 보이는 바와 같이 학습은 모든 경우를 정확하게 담당할 수 있는 완전한 행위를 구현하지는 못할 것이다.

$$\bigcup_{i=0}^{|B|-1} S_i \neq S \tag{5}$$

이렇게 $\beta(S) \mapsto A$에 포함되지 않아 코딩되지 않은 센서 패턴에 대한 해답은 단순히 위의 설계 단계 1.2.2 를 수행하는 것이다. 이미 행위의 개수가 포화되었기 때문에 이 경우의 해답은 거의 즉시, 최악의 경우에도 $|B|$ 번의 시도 후에 찾아질 수 있다.

다음으로 학습의 단계에서 행위의 개수가 포화되었다고 판단한 후에도 어떤 경우에는 또다시 새로운 행위가 생성되어야 할 조건 즉, 설계 단계 1.2.3의 조건이 나타날 수도 있다. 이런 경우에는 다시 1.2.3의 단계를 수행하여 재학습을 행하여 행위를 추가하고, 필요한 경우 다시 행위의 집합을 재코드화하면된다. 이 방법을 통하여 로봇이 동작 하면서 그 행위를 향상(update)하게 한다. 여기서 주 목할 것은 하나의 상황을 이루는 센서 패턴 벡터의 수 $|S_i|$ 나 생성되는 행위의 수 $|B|$ 는 무한정 증가하지 않는 다는 것이다. 이는 $|S_i|$ 나 $|B|$ 가 증가함에 따라 코드화되지 않은 센서 패턴의 개수가 다시 줄어들기 때문이다.

위의 재학습 과정은 생물체가 태어나 경험을 많이 해나가면서 보다 세련되고, 합리적인 행위를 해 나가게 되는 것과 같은 것이다. 또한 이는 발생 빈도 수가 극히 낮은 상황을 처음부터 고려하여 설계에 반영하는데 따르는 비용 부담을 줄이는 효과가 있기도 하다.

III. 실험

1. 실험환경

위에서 제시한 행위의 자동설계 방법을 이동 로봇의 2차원 주행에 적용하여 그 유효성을 검증하였다. 사용한 로봇(Magellan Pro™)은 원통형의 측면에 16개의 초음파 센서가 22.5° 간격으로 배치되어 주변의 물체와의 거리를 감지한다. 따라서 본 실험에서의 상황은 16개의 초음파 센서의 값에 의해서 결정되게 된다. 실험의 편의상 (6)과 같이 각 초음파 센서의 출력은 한 샘플링 시간 동안에 로봇이 이동할 수 있는 최대 거리인 1.5m를 기준으로 물체가 그 보다 가깝게 있으면 '1'(ON)로, 멀리 있으면 '0'(Off)으로 하여 두 가지의 2진(binary)값만을 가지도록 하였다.

$$u_i \begin{cases} 0, & d_i > d_{\max},\ i=0,1,2,\cdots,15 \\ 1, & d_i \le d_{\max},\ i=0,1,2,\cdots,15 \end{cases} \tag{6}$$

따라서 로봇이 느낄 수 있는 총 센서 패턴은 65536($=2^{16}$) 경우가 된다. 주행을 하는 주변 환경은 시작점과 목표점 사이에 제한된 크기의 장애물이 임의로 배치된 직사각형의 넓은 방으로 설정하였다. 그림 2와 3에 실험에 사용된 로봇과 환경을 각각 나타내었다.

로봇에게는 장애물에 부딪히는 것은 나쁘다는 것과, 목표점으로 이끌리는 성향을 본능으로 주었다. 또한 실험의 진행을 위하여 동물에 비유하자면 먹이의 냄새와 같이 임의의 위치에서 목표점의 방향을 알 수 있게 하였다.-이는 실제로는 광학엔코더(optical encoder)를 이용하여 주행거리계(odometry)를 계산하여 얻는다.- 학습 단계 n번째 시간에서 로봇에게 주어지는 보답은 주어진 본능에 따라

$$r_a(s) = G_a(p_n - p_{n-1}) \cdot g/|g| - G_r\zeta \tag{7}$$

와 같이 주어졌다. 여기서 p_n은 출발점을 원점으로 할 때의 n번째 샘플링 시각에서의 로봇의 위치 벡터이고 g는 목표점의 위치 벡터를, G_a와 G_r은 상수로서 각각 끌림 이득(attractive gain)과 반발 이득(repulsive gain)을 나타내며, ζ는 장애물과의 충돌을 나타내는 값으로 충돌 시에 '1'을 충돌하지 않았을 때는 '0'값을 가진다. 위 (7)의 첫번째 항의 내적은 목표지점의 방향으로의 전진 정도가 클 수록 보답이 크게 하

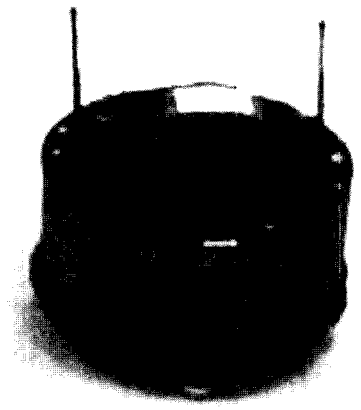

그림 2. Magellan Pro™.
Fig. 2. Magellan Pro™.

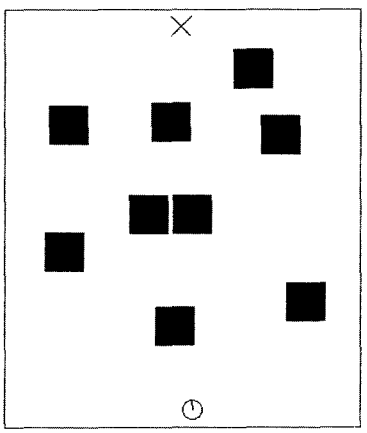

그림 3. 실험 환경의 예.
Fig. 3. An environment for experiment.

고, 두 번째 항은 장애물과 충돌하지 않을 수록 보답이 커지게 하는 역할을 한다. 한편 각 행위의 동작은 주어진 상황에서의 속도 벡터로 결정된다. 속도 벡터는 로봇 자신의 자세를 기준으로(ego-centric)한 형태로 속도 벡터의 방향 ϕ와 속력 ρ로 구성되고 전체 가능한 동작 공간 A는 방향을 5° 간격으로, 속력을 0.1m/s간격으로 변화시켜 가면서 구성하였다. 학습은 시작점에서 출발하여 목표점까지 도달하는 한 번의 에피소드(episode)가 끝나면 다시 시작점과 목표점의 위치를 다양하게 변화시켜가면서 여러 번의 에피소드를 진행 시켰다. 학습이 끝난 후 학습된 데이터는 다층퍼셉트론(multi-layer perceptron) 신경망 구조와 2진 논리연산의 두 가지 형태로 구현하였으며 두 형태 모두 같은 결과를 보였다. 본 실험에서의 신경망의 입력 노드는 16개의 초음파센서의 출력으로 그 값은 2진(binary) 값이다. 출력 노드는 속도벡터의 방향과 크기이다. 2진 논리연산의 형태로 구현한 경우는 특히 행위의 구현형태가 간단해지고, 하드웨어 이식성(hardware targettability)이 높아지며 반응속도가 상당히 빨라지는 강점을 가진다. 이 방법은 2진 값을 갖는 16개의 센서 값을 각각 $u_0, u_1, \cdots, u_{15}$라 할 때, 한 센서 패턴은 감지된 값이 '1'인 경우의 u_i와 '0'인 경우의 $u'_i(i=0,1, ...,15)$의 곱셈항(product term)으로 표현된다. 한 행위는 해당 상황을 구성하는 다중 센서 패턴 벡터들로 이루어져 있으므로 결국 위의 곱셈항들의 boolean 합으로 표현된다. 즉, 다시 말해 한 행위는 2진 논리 연산의 곱의 합(sum of product(SOP)) 형태로 표현되며 이 때 각각의 곱셈항이 바로 그 상황을 구성하는 센서패턴 벡터들이다. 행위가 일단 2진 논리 연산의 형태로 표현된 이상 물론 여러 가지 방법을 이용하여 간략화 하여 보다 간결한 형태로 행위를 인코딩 할 수 있다[10]. 다음의 (8)은 2진 논리 연산으로 구현된 행위의 한 예이다.

$$b_0 = (u_0 u'_3 u_{10} + u_3 u_5 u'_6 u_{12} u'_{13} u_{15} + u_0 u_7 u_6)[\phi_0 \rho_0]^T \quad (8)$$

2. 실험 결과

이동 로봇의 주행에 강화 학습을 통한 자동 행위 설계 방법을 적용한 결과 보통 3~6개의 행위가 만들어 졌으며, 이는 에피소드가 진행되는 동안 많이 증가하지 않았다. 그림 4에 여러 가지로 다르게 장애물을 배치한 환경에서의 에피소드와 행위의 생성 개수의 관계를 나타내었다. 이 결과는 어떤 환경과 로봇이 주어졌을 때, 로봇의 센서가 느끼는 패턴 중에 유사한 패턴은 동일 상황으로 통합이 되었으므로 환경의 의미 있는 단위의 상황-예를 들자면 '전방에 장애물 존재'상황-의 개수는 유한한 값이 되고, 따라서 이에 대응되는 행위의 개수가 에피소드가 증가함에 따라 무한히 증가하지 않고 일정 값에서 포화 될 것이라는 예측과 일치하는 것이다.

이로서 실제 수많은-본 실험에서는 65536(=216)- 센서 패턴을 감당할 수 있는 몇 개의 행위를 생성할 수 있음을 확인하였다. 표 1에 r_{th}=0.15, r_{ch}=0.05의 경우에 설계된 대표적인 행위들의 예를 나타내었다.

표 1의 행위들을 기존의 설계자의 입장에서 해석한다면 각각 "제자리에서 회전", "직진", "장애물을 회피하면서 전진"에 해당하는 행위임을 볼 수 있다. 이 들의 예를 통하여 설계자가 설계한 것과 비슷한 경향의 이성적인 설계가 이루어 진 것을 볼 수 있다. 학습을 통하여 스스로 설계된 행위를 가지고 로봇의 주행을 시킨 결과 성공적인 장애물 회피와 목표점으로의 도착 임무를 수행하였다.

IV. 결론

본 논문에서는 로봇에게 최소한의 본능만을 준 상태에서 심각한 보답의 변화를 기준으로 삼아, 스스로 필요한 행위를 설계하는 방법론을 제안하였다. 이 방법의 핵심 결과는 로봇 스스로 필요한 행위의 개수와 종류를 결정하고 설계할 수 있다는 것이다. 본문에서 기술한 특징 외에도 이 방법은 근본적으로 로봇이 실제 느끼는 센서 값을 근간으로 한(sensor-based) 방법이므로 설계된 행위는 센서 패턴의 직접적인(explicit) 함수의 형태로 표현되어 별도의 다른 처리 과정 없이 즉각 반사적으로 동작을 제시해 준다. 이것은 바로 센서와 동작이 긴밀하게 결합된 지각 모터(sensory motor)로 행위를 구현하는 일반적인 행위 기반 로봇의 형식을 충실이 따르는 형태이다. 또한 각 행위의 생성에서의 병렬처리 특징과 행위를 첨가하여 나가는 설계 방식 역시 행위 기반 로봇의 장점을 잘 살린 것이다. 특히 행위를 2진 논리 연산으로 구현하는 경우에는 단순성, 속도, 하드웨어 구현의 측면에서 특별한 강점을 갖는다. 또한 학습에 사용되는 상황이 특정한 모양의 환경의 전체 형상이 아니고, 로봇의 센서 입장에서 느끼고 분류되어(classified) 추출된 요소(elementary) 형상이

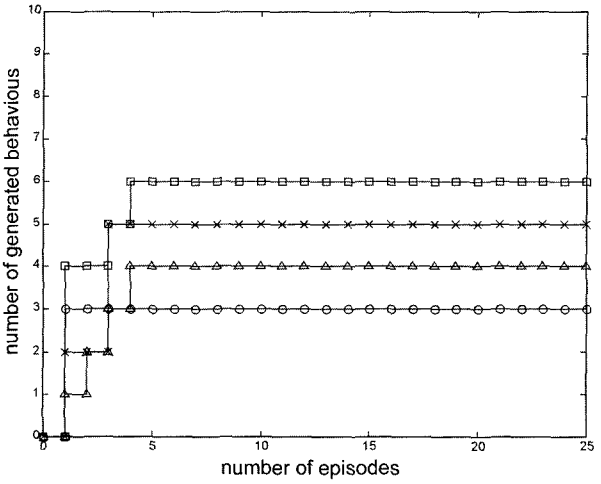

그림 4. 여러 가지 환경에서의 에피소드의 증가에 따른 생성된 행위의 개수.

Fig. 4. Resultant number of generated behaviors versus number of episodes in different arrangements of obstacles.

표 1. 실험 결과의 예.

Table 1. Some examples of experiments.

대표적 sensor pattern	ϕ[deg]	ρ[—]
$s = [1111111111100011]^T$	-85	0.1
$s = [0000111111111111]^T$	5	0.5
$s = [0000000001111100]^T$	50	0.3

므로 전체적인 환경의 모양과 관계없이 다른 환경에서도 동작하는 장점을 자연스럽게 가지게 된다. 다만, 학습 시에 경험하지 않은 센서 패턴에 대한 문제가 있으나 이는 확률적으로는 크게 문제되지 않는 것이고, 기술한 방법대로 지속적인 재학습을 통하여 해결이 가능하다. 본 논문에서는 반사 제어(reactive control)방식의 전형적인 문제점인 로봇의 지엽적인 시야에서 오는 지역최소값(local minima)[8], 기억 장치를 사용하지 않을 때의 반복 행위(cyclic behavior)[9]의 문제를 다루지 않았다. 이 문제는 본 연구의 주된 관심사인 개별적인 행위의 설계의 문제와는 별개의 문제로 볼 수 있는 조정의 문제로서 별도의 전역적인 정보(global information)와 기억 장치가 제공되지 않으면 해결되기 어려운 문제이다. 본 논문은 근본적으로 경쟁적인 조정(competitive coordination) 방식에 의한 조정 방식을 채택하고 있는 상황이며 물론 위의 문제점이 존재한다. 이는 본 논문의 연구 범위를 벗어난 분야이며 앞으로 계속 연구를 진행할 주제이다.

참 고 문 헌

[1] R.C. Arkin, *Behavior-Based Robotics,* The MIT Press, Cambridge, 1998.

[2] R.A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robotics and Automation,* vol. RA-2, no. 1, pp. 14-23. 1986.

[3] R.C. Arkin, "Towards cosmopolitan robots : Intelligent navigation in extended man-made environments," *Ph.D. Dissertation, COINS Tech. Rpt., 87-80, Univ. of Massachusetts, Dept. of Computer and Information Science,* pp. 143-177, 1987.

[4] P. Maes, "The dynamics of action selection," *Proc. Int. Joint Conf. on Artificial Intelligence,* Detroit, MI, pp. 991-997, 1989.

[5] A. Saffiotti, K. Konolige, and E. Ruspini, "A multivalued logic approach to integrating planning and control," *Artificial Intelligence* 76, pp. 481-526, 1995.

[6] R.J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning,* 8, pp. 229-256, 1992.

[7] C. J. C. H. Watkins, and P. Dayan, "Technical note Q-learning," *Machine Learning,* 8, pp. 279-292, 1992.

[8] Y. Koren and J. Borenstein, "Potential field methods and their inherent limitations for mobile robot navigation," *Proc. IEEE Int. Conf. Robotics and Automation,* Sacramento, CA, pp. 1398-1404, 1991.

[9] A. Mukerjee and A. D. Mali, "Reactive robots and amnesics : A comparative study in memoryless behavior," *IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol: 292 , pp. 216 -226, May 1999.

[10] D. A. Pucknell, *Fundamentals of Digital Logic Design with VLSI Circuit Applications,* Prentice Hall of Australia Pty Ltd, Sydney, Australia, 1990.

윤 도 영

1970년 12월 30일생. 1995년 서강대학교 전자공학과 (공학사). 1997년 서강대학교 전자공학과 (공학석사). 1999년~현재 고려대학교 전기공학과 박사과정 수료, KIST 학생연구원. 관심분야는 system implementation, control theory application, robot control architecture.

박 귀 태

1947년 10월 25일생. 1975년 고려대학교 전기공학과 (공학사). 1977년 고려대학교 전기공학과 (공학석사). 1981년 고려대학교 대학원 전기공학과 (공학박사). 1981년~현재 고려대학교 전기공학과 교수. 현 IBS Korea 회장. 관심분야는 intelligent controls, image processing, mobile robotics, home automation, intelligent building system(IBS).

오 상 록

1958년 6월 7일생. 1980년 서울대학교 전자공학과 (공학사). 1982년 한국과학기술원 전기 및 전자공학과 (공학석사). 1987년 한국과학기술원 전기 및 전자공학과 (공학박사). 1988년~현재 KIST 책임연구원. 관심분야는 robotics and automation, intelligent control and machine lear- ning, advanced Control system theory.