Proceedings of the 13th
KACC, October 1998

퍽인홀 작업제어에 적용되는 다중 목표를 갖는 영역기반 Q학습법

A Region-based Q-learning method with multiple goals applying to a peg-in-hole task control

°차 승 현*,**, 서 일 홍*, 조 영 조**, 김 재 현*, 박 정 민**

* 한양대학교 전자공학과(Tel : +82-345-408-5802; Fax : +82-345-408-5803 ; E-mail: shcha1@amadeus.kist.re.kr)
** 한국과학기술연구원 지능제어연구센터(Tel : +82-2-958-5759; Fax : +82-2-958-5749)

Abstract : A new learning control method which uses a region based Q-learning algorithm with multiple goals is proposed. RQ-learning algorithms enable robot manipulator to more smoothly in real environments with the ability to learn its actions fast. It is well known that Jam/Wedge states have to be avoided in achieving peg-in-hole task goals. This paper presents a method to achieve the given task goal as well as to avoid Jam/Wedge states by using a RQ-learning algorithm. Simulation results assure the validity of the proposed method

Keywords : Q-learning, RQ-learning, Multiple Goal, Peg-in-hole, Behavior learning

1. 서론

로봇의 사용목적가운데 가장 큰 분야는 조립작업이다. 많은 로봇이 산업현장에서 조립작업을 위해 사용되고 있으며 점점 인간의 영역까지 확대되어 가고 있다. 따라서 퍽인홀 작업을 통하여 조립작업에서 흔히 발생되는 문제점인 Jam/Wedge[6], 이산사건식 접근방법을 이용하여 상태인식을 통해 작업을 제어하는 연구가 꾸준히 있다[1,3,7].

본 논문은 로봇이 학습 과정을 통하여 3차원 공간에서 임의의 홀에 퍽을 넣는 작업 방법을 제안한다. 그리고 기존의 Q학습법의 불연속적인 성질을 보완하는 연구들이 있었는데[2,4], 그 중에 Q학습법에 비해 기억공간을 줄여주고 학습시간을 단축시켜줄 뿐만 아니라 실제 로봇의 움직임을 좀 더 부드럽게 해주는 영역기반 Q학습법(RQ-learning)을 사용하였다[2].

퍽인홀 작업에 있어서 Jam이나 Wedge상태는 작업을 더 이상 진행시킬 수 없는 상태가 될 수 있으므로 피해야 한다. 따라서 Goal과 Jam/Wedge상태를 독립적인 모듈로 두어 각각 병렬적으로 학습을 하게 하는 multiple goal 학습방법을 적용하였다[5]. 이때 main goal을 얻었을 경우 보답을 증가시키고, Jam/Wedge가 발생했을 경우 보답을 감소시켜서 로봇이 환경으로 얻은 현재 상태에서 조정자에 의해 행동을 얻게 되며 최종적으로 로봇이 Jam과 Wedge상태를 거치지 않고 main goal을 얻도록 하는데 목적이 있다.

2. Q-learning과 RQ-learning

2.1 Q-learning

강화 학습(Reinforcement learning) 방법들은 자율적인 로봇과 환경사이에 이산 시간(discrete time)의 주기적인 반복으로 모델링 될 수 있다. 반복 과정은 다음과 같다. 먼저 로봇이 현재 상태를 파악하고 그 상태에서 행동을 수행한다. 바뀐 환경은 다음 상태를 만들어내고 로봇은 특별한 행동에 대하여 보상을 받게되고 이러한 보상에 기반을 두어 로봇은 그 상태에서 행해지는 행동에 대해 평가를 하게된다. Q-learning은 로봇(agent)의 행동의 순서를 평가함으로써 최적의 행위를 할 수 있는 능력을 주며, Q-value는 로봇이 특정한 상태에 대하여 적절한 행동을 선택하게 해준다. Q학습법의 알고리즘을 간략하게 소개하면 다음과 같다.

식(1)에서 현재 상태에서 가장 큰 Q값을 가지는 행위를 얻게되고, 다음 상태에서 얻은 보답을 식(2)를 통해서 Q값을 갱신시켜준다. 이러한 작업을 반복함으로써 로봇은 가장 큰 Q값을 가지는 행위를 계속 취함으로써 goal을 얻게된다.

$$f_i \leftarrow a \quad \text{such that} \quad Q_i^a(t+1) = \max_{b \in A}\{Q_i^b(t)\} \tag{1}$$

$$Q_i^a(t+1) = \alpha Q_i^a(t) + (1-\alpha)(r_i^a + \gamma \max_{b \in A}\{Q_{i+1}^b(t)\}) \tag{2}$$

,i는 현재 상태, f_i는 책략표에 따른 현재 상태에서의 행동, A는 가능한 행동, $Q_i^a(t+1)$는 다음 반복때의 현재 상태 I에서의 행동 a의 Q값이다. α(0< α<1)는 learning rate이고, γ는 감쇠요소이다.

2.2 영역 기반 Q-learning(RQ-learning)

2.1에서 보여준 기존의 Q-learning은 로봇이 실제 환경에서 모든 상태에 대하여 최적의 행위를 배우기 위해서 너무 많은 기억공간과 시간을 요한다. 더욱이 불연속적인 상태공간 안에서 불연속적인 행동을 발생시키기 때문에 로봇이 부드러운 동작을 수행할 수 없다. 영역기반 Q-learning은 point-wise Q-learning의 확장이라고 볼 수 있다.[1] 기존의 Q-learning과는 달리, RQ-learning은 상태공간 안의 모든 상태들에 대해서 학습할 필요가 없다. 사실상 단지 몇 개의 대표적인 상태들에 대해서만 행동을 최적화 하는 것이 필요하다. 따라서 상태분해능에 따라 주변상태를 정의하고 현재의 상태에서 받은 보답을 거리에 따른 효과함수에 따라 주변상태로 전파시켜준다(식 3).

$$r_j = \mu_{i,j} r \qquad Q_{i,j}^{a_i} = \sum_{n=0}^{\infty} \gamma^n \mu_{i+n,j} r_{i+n} \tag{3}$$

,i는 현재 상태, j는 주변 상태, μ는 효과함수이다.

RQ학습에서는 연속된 상태와 행위공간에서 학습을 하기 때문에

Q학습에서 사용한 행위값표 대신에 특정 형태로 모델링하는 것이 필요하다. 따라서 특정 행위에서 최고치를 갖고 특정행위와 관계가 멀수록 일정하게 행위값이 작아지는 삼각 형태의 행위값 모델링을 한다. 그래서 주변상태의 행위값 모델로부터 현재상태의 최적행위를 생성해낸다.

3. Multiple goal task

로봇이 일련의 목표들을 행하기 위해서는 2장에서 나열한 알고리즘을 확장할 필요가 있다. 로봇이 수행해야할 목표가 n개, Γ^1, Γ^2,..., Γ^n 있다고 가정하면 각 목표 Γ^i는 보답 함수 R^i를 가진다. 그리고 각각의 목표는 시간에 따라 활동 상태 g_t^i를 가지며 모든 목표들에 있어서 활동 벡터를 가진다.

$$\bar{g} = g^1 \cdot g^2 \cdot g^3 \cdots g^n \tag{4}$$

새로운 보답 함수는 현재의 전체 상태와 현재의 활동 벡터에 의존한다.

$$R(x,\bar{g}) = \sum_{i=1}^{n} R^i(x) g^i \tag{5}$$

목표가 활동성이면 그 목표를 얻을 때까지 활동성이 지속되며, 로봇이 하나의 목표를 달성하면 그 목표의 활동성은 없어지고 다른 활동성을 가진 목표를 추구한다. 이러한 다중 목표 작업들은 Markov decision processes로서 모델링이 되어질 수 있음을 보였다[5].

4. 학습을 통한 Peg-in-Hole 작업

펙인홀 작업에 있어서 펙과 홀사이 가해지는 힘과 반발력사이의 관계, 펙을 홀에 삽입할 때 생기는 Jam과 Wedge의 상태를 해결하기 위하여 많은 연구들이 있었다. 본 논문에서는 3장에서 언급한 것과 같이 goal을 펙인홀 작업에 있어서 펙이 홀에 완전히 들어간 상태뿐만 아니라 작업을 더 이상 진행시키기 어려운 상태, 즉 Jam과 Wedge상태를 또 다른 중간 목표로 잡아 그 상태를 피하여 최종 목표에 도달하도록 하는데 목적이 있다. 본 논문에서는 5자유도를 가진 로봇과 원형 펙, 홀을 대상으로 하였다.

4.1 상태의 정의와 주변 상태

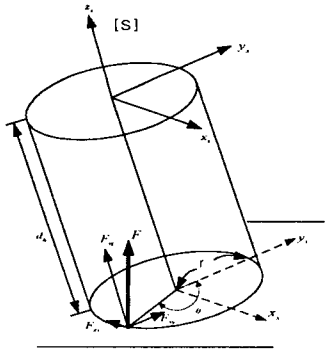

그림 1. 펙의 접촉으로 얻어지는 힘 벡터
Fig. 1. Force vector

3차원 공간상에서 홀과 펙이 접촉할 때 생기는 힘을 기반으로 상태 공간을 형성한다. 접촉시 생기는 힘 벡터를 일정한 크기의 구로 정규화시킨 벡터를 상태로 정의하고 상태 공간에서의 행동은 자유도가 5축인 로봇과 원형 펙을 대상을 하기 때문에 x, y, z, pitch의 이동으로 정의한다. 상태 공간은 각 상태 벡터를 x, y, z로 정사형시켰을 때 나오는 값을 n개의 그룹으로 나누어 구분할 수 있다.

RQ-learning에서는 3차원 공간상의 단위구로 향하는 모든 벡터가 상태가 되며, 각 상태는 구면을 구성하는 4개의 주변 상태를 가지고 있으며 현재 상태와 주변 상태와의 관계는 거리에 따른 효과

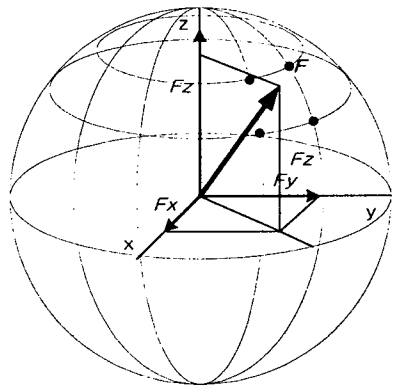

그림 2. 정규화시킨 힘 벡터 와 주변 상태들
Fig. 2. normalized force vector and neighboring states

함수에 따라 현재 상태에서 받은 보답을 주변 상태로 할당한다(그림 2).

4.2 Multiple Goal을 가지는 Modular 구조

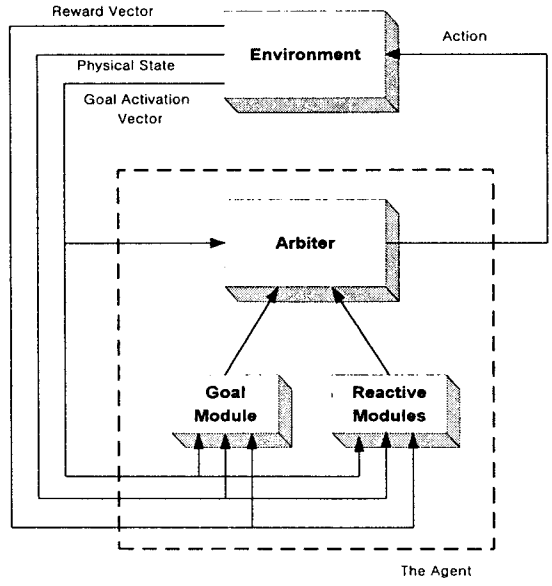

그림 3. 다중 모듈 구조
Fig. 3. Modular Architecture

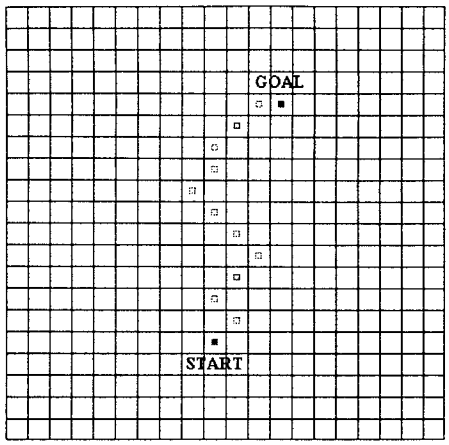

그림 4. 한 개의 goal을 가진 Q학습법
평균적으로 *6000번의 time step* 후에 학습이
제대로 됨
Fig. 4. Q-learning that has one goal

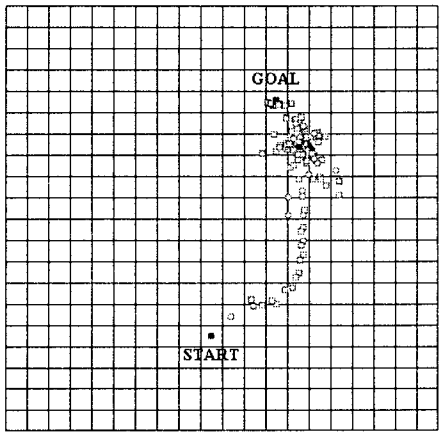

그림 5. 한 개의 goal을 가진 RQ학습법
평균적으로 *5000번의 time step* 후에
학습이 제대로 됨
Fig. 5. RQ-learning that has one goal

펙인홀 작업에서 다중 goal을 가지는 Modular Architecture는 최종 Goal에 해당하는 Goal Module과 Jam 상태에 해당하는 Reactive Module, 2개의 module을 가진다.(그림 3)

여기서 Agent는 환경으로부터 얻은 현재 상태에서 Goal Module과 Reactive Module에 대해 병렬적으로 학습을 한다. 이때 Goal Module에서는 양의 보답을 주고, Reactive Module에서는 음의 보답을 준다. 따라서 현재 상태에서 로봇은 Goal을 향해 가려는 행동과, Jam 상태에 빠지지 않으려는 행동을 가지게 되는데, 식(5)과 같이 조정자는 두 module이 가지고 있는 Q값을 비교하여 큰 값을 가지는 행동을 취하게된다.

$$f_m(x,\bar{g}) = \arg\max_{a\in A} Q_m(x,\bar{g},a),$$
where,
$$Q_m(x,\bar{g},a) = \max_{i\in[1\ldots n]}\left[Q^i(x,a)\cdot g^i\right] \quad (6)$$

여기서 g는 3장에서 언급한 activation vector이고, a는 action, x는 현재 상태의 주변 상태이다.

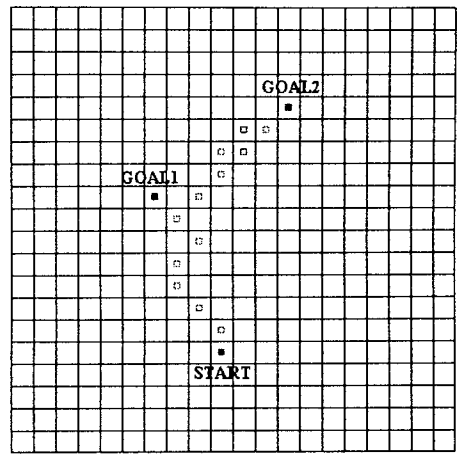

그림 6. 두 개의 goal을 가진 Q학습법
평균적으로 *5000번의 time step* 후에 학습이
제대로 됨
Fig. 6. Q-learning that has two goal

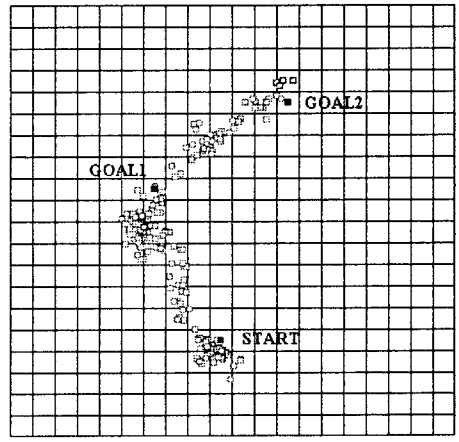

그림 7. 두 개의 goal을 가진 RQ학습법
평균적으로 *3500번의 time step* 후에 학습이
제대로 됨
Fig. 7. RQ-learning that has two goal

action은 x, y, z, pitch 4방향이 있는데 각각 삼각형 행위값 모델을 가지며 가장 큰 값을 가지는 각각 4개 벡터의 합벡터가 행위를 결정한다.

5. Simulation

우선 goal 하나인 작업에서 Q학습법과 RQ학습법을 도입하여 두 가지 방법의 성능을 테스트했을 경우, 그림 4와 그림 5에서와 같이 Q학습법에 비해 RQ학습법이 학습시간에 소요되는 시간이 적게 드는 것을 볼 수 있다.

또 2개이상의 goal을 가지는 multiple goal task에서는 병렬적으로 두 개의 goal을 동시에 학습하기 때문에 하나의 goal을 가질 때보다 time step이 두배가 걸리지 않음을 알 수 있고, 여기서도 그림 6, 그림 7와 같이 RQ학습법의 사용이 더욱더 효과가 좋음을 알 수 있었다.

펙인홀 작업의 시뮬레이션에서는 3차원 공간을 x, z 2차원 공간

으로 축소하여 테스트하였으며 그에 따라 상태공간, 행위들이 줄어들었다. 펙이 홀을 찾아 근처까지 오는 것은 시각정보를 이용하였다고 가정하고 펙이 홀 주위에 닿았을 때부터 학습을 시작였고, Jamming 상태는 그림 8과 같이 x, z, theta방향으로 어떠한 움직임도 취할 수 없는 상태로 정의하였다. 시뮬레이션 진행 과정을 그림 9, 그림 10, 그림 11에서 나타내었다.

6. 결론 및 추후과제

본 논문에서는 펙인홀과 같은 조립작업을 학습을 통해 하고자 하는 목적에서 시작하여 영역기반 Q학습법으로 펙인홀 작업이 수행됨을 시뮬레이션을 통하여 보였다. 앞으로 5자유도를 가진 로봇팔을 이용하여 3차원 공간상에서의 펙인홀 작업 방법을 검증할 것이다. 그리고 현재는 Jam 상태만을 Reactive Module로 두었으나, 홀을 찾는 과정에서의 장애물을 피하는 과정도 연구할 필요가 있다.

참고문헌

[1] D.Austin and B.J.McCarragher, "Experiment in Force Controlled Assembly suing Discrete Event Framework," IEEE International conference on Intelligent Robotics and System, pp.668-674, Aug. 1997.

[2] J.H.Kim, I.H.Suh, S.R.Oh, Y.J.Cho and Y.K.Chung "Region Based Q-learning using Convex Clustering Approach" Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), pp.601-607, 1997.

[3] B.J.McCarragher, "Robotic Assembly and Trajectory Planning using Discrete Event Modeling", Second IEEE Workshop on Emerging Technologies and Factory Automation, 1993

[4] A. Moore "The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces," Advances in Neural Information Processing Systems 6, pp.711-718.

[5] S. Whitehead, J. Karlsson and Josh Tenenberg "Learning multiple goal behavior via task decomposition and dynamic policy merging," *Robot Learning* : Kluwer Academic Publishers, pp45-78

[6] D.E.Whitney, "Quasi-static Assembly of Compliantly Supported Rigid Body", ASME journal of Dynamic System, Measurement and Control, Vol.104, pp65-77, 1982

[7] 전은석, 유범재, 조영조, 윤태웅, "이산사건식 접근방법을 통한 지능형 로봇의 작업계획 및 제어," 대한 전기학회 하계학술대회 논문집 , 1997.

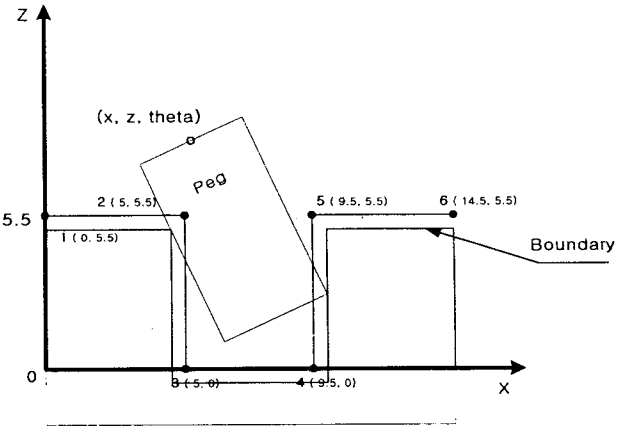

그림 8 홀, 펙의 모델링과 Jam상태
Fig. 8. the modeling of hole and peg and Jamming

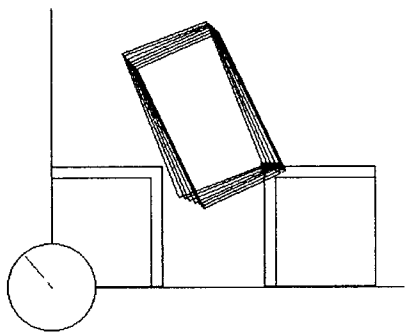

그림 9. 시작 단계
Fig 9. Start

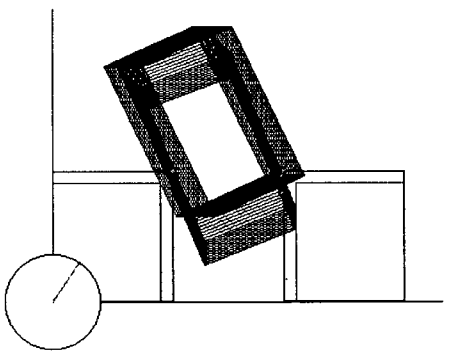

그림 10. Jam 상태
Fig 10. Jam state

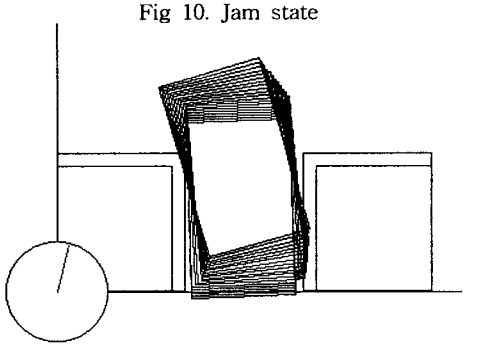

그림 11. Goal 상태
Fig. 11. Goal state