이동 로봇의 강인 행동 계획 방법

A Robust Behavior Planning technique for Mobile Robots

이 상 형[1] · 이 상 훈[2] · 서 일 홍[†]

SangHyoung Lee[1] · Sanghoon Lee[2] · Il Hong Suh[†]

Abstract We propose a planning algorithm to automatically generate a robust behavior plan (RBP) with which mobile robots can achieve their task goal from any initial states under dynamically changing environments. For this, task description space (TDS) is formulated, where a redundant task configuration space and simulation model of physical space are employed. Successful task episodes are collected, where A* algorithm is employed. Interesting TDS state vectors are extracted, where occurrence frequency is used. Clusters of TDS state vectors are found by using state transition tuples and features of state transition tuples. From these operations, characteristics of successfully performed tasks by a simulator are abstracted and generalized. Then, a robust behavior plan is constructed as an ordered tree structure, where nodes of the tree are represented by attentive TDS state vector of each cluster. The validity of our method is tested by real robot's experimentation for a box-pushing-into-a-goal task.

Keywords : Robust Behavior Planning(RBP), Planning, Reactive plan

1. 서 론

이동 로봇의 서비스를 위한 행동 계획은 태스크를 성취하기 위한 일련의 행동 순서로 볼 수 있다. 행동 계획은 일반적으로 계획기(planner) 또는 사람이 작성한다. 특히 사람이 직접 작성한 행동 계획을 일반적으로 프로그램이라고 부른다. 산업용 로봇의 경우처럼 작업 환경이 로봇 중심으로 고정되어 있고, 수행하는 태스크가 고정적이라면 로봇의 행동 순서를 사람이 직접 프로그램 하기 쉽다. 그러나 서비스 로봇(이동 로봇)은 작업 환경이 인간 중심의 동적인 환경이기 때문에 사전에 프로그램 된 행동 계획으로 태스크를 성취하기 어렵다[8].

로봇이 수행하기 위한 행동 순서를 프로그램 하는 다른 방법으로 행동 계획기를 사용한다. 일반적인 행동 계획기는 로봇의 현재 상태에서 태스크 성공 상태까지의 행동 순서(또는 경로)를 작성한다. 그러나 일반적인 행동 계획기가 작성한 행동 계획은 행동 계획이 수행되는 도중에 예상하지 못한 환경 변화에 따른 예외 상황이 발생하면 실패할 수 있다. 이를 보완하기 위한 행동 계획 방법으로 conditional planning, re-planning, continuous planning[9][10] 등이 제안되어 사용되고 있다. 이들 계획 방법들은 예외 상황 발생 기회를 줄이거나 계획 실패 시 시스템적으로 복구하는 방법이다. 하지만 동적 상황 변화에 따른 예외 상황이 발생했을 때 현재 상황에서 목표까지 계획을 작성하기 때문에 동적인 환경에서 예외 상황이 발생하면 추가적인 비용(계획 실패 또는 재계획)이 발생한다.

Bryson은 정확한 단계별 순서를 미리 결정할 수 없는 환경에서 사용할 수 있는 반응 계획(reactive plan)을 제안했다. 이 계획 방법은 우선 순위(priorities), 전제 조건(preconditions), 행동(actions)으로 구성되어 있다[1]. Bryson이 제안한 반응 계획은 주변 환경에 대해 완벽히 이해 가능하다는 가정을 바탕으로 단순한 실험 환경에 대해 프로그램 및 테스트되었다. 다시 말해, 로봇이 취할 수 있는 단위 행동들과 그에 속한 주목할 만한 상태들을 이미 알고 있다고 가정하였다. 하지만 태스크와 관련된 단위 행동들과 성공적으로 태스크를 수행한 몇 개의 상태-행동 쌍의 순서(또는 task-related trajectory[12])로부터 주목할 만한 상태를 완벽히 이해하는 것은 매우 어렵고 반응 계획을 프로그램 하기 위한 방법을 완벽히 제시하지는 못하였다.

본 논문에서는 성공적으로 태스크를 수행한 에피소드(또는 경로) 라이브러리를 통해 반응 계획을 생성하

※ 본 연구는 정보통신부에서 수행되고 있는 'URC 를 위한 컴포넌트 기술 개발 및 표준화' 과제의 지원을 받아 수행되었습니다.

[1] 한양대학교 정보통신대학 석사과정(zelog@incorl.hanyang.ac.kr)
[2] 한양대 지능 및 상호작용연구실 연구교수 (shlee@incorl.hanyang.ac.kr)
[†] 교신저자 : 한양대학교 정보통신대학 교수(ihsuh@hanyang.ac.kr)

는 방법을 제안한다. 에피소드(또는 경로) 라이브러리[4]는 RRT[5], A*[7]와 같은 경로 계획기(path planner)를 통해 태스크를 성공적으로 수행할 수 있는 에피소드(또는 경로)를 포함하도록 수동적으로 설계하는 것이 가능하다. 여기서 반응 계획은 에피소드 라이브러리로부터 몇 개의 주목할 만한 상태들을 추출하는 과정과 그 상태들과 관련된 행동들의 집합을 구성하는 과정을 통해 생성된다. 이 때, 각각의 주목할 만한 상태들은 행동들로 구성된 집합을 포함하고 있고, 순서가 있는 트리 구조로 구성된다. 그리고, 에피소드 라이브러리를 이용해 로봇이 예상치 못한 상태 변화에 대해 행동하도록 설계되어 있어 강인한 계획이라고 말할 수 있다. 이와 같은 특징 때문에 앞으로 본 논문에서 제안한 계획 방법을 강인 행동 계획(Robust Behavior Plan - RBP)이라고 부르도록 하겠다.

경로 라이브러리 방법[12]은 현재 상황에서 행동하기 위한 상태들의 할당을 조정하는 방법이다. [12]에서 라이브러리의 경로는 kd-tree가 적용된 경로 라이브러리에서 현재 상태와 가장 가까운 이웃 경로를 찾아내는 과정을 통해 생성된다. 따라서 'if-then'으로 경로를 표현하는 것이 어렵다.

강인 행동 계획은 태스크 에피소드가 태스크의 하위 목표들로 표현되는 것이 가능하도록 주목할 만한 상태들을 추출하는 것에 관심을 갖는다. 강인 행동 계획의 주된 아이디어는 다음과 같다. 현재 상태와 가장 가까운 하위 목표를 성취하고, 그 다음 더 높은 하위 목표를 성취하는 과정을 통해 태스크의 최종 목표를 달성한다. 이 방법은 현재 로봇의 수행 가능한 행동이 명확하지 않은 상태에서 로봇이 지니고 있는 명의적 계획(nominal plan)들 중에서 현재 상태와 가장 가까운 계획을 수행하기 때문에 지역적 최대치(local maxima)에 빠지는 것을 피할 수 있도록 한다. 이것은 [11]에서 언급하고 있는 자기 목적적인 원칙(autotelic principle)과 유사하다. 이 원칙의 주된 아이디어는 새로운 자극에 대한 흐름 원칙에 따라 현재 자신과 가장 가까운 목표를 수행하기 위해 새로운 하위 목표를 구성하는 것이다. 그러나, [11]에는 아이디어의 정당성을 보이기 위한 상세한 실험 환경에 대해 언급되어 있지 않다.

본 논문의 구성은 다음과 같다. 절 2에서 강인 행동 계획의 생성 방법에 관해 논의한다. 절 3에서는 Box-Pushing-into-a-Goal 태스크에 강인 행동 계획을 적용한 시뮬레이션과 실험 결과를 보여준다. 마지막으로 절 4에서 요약과 앞으로 할 일에 대해 논의한다.

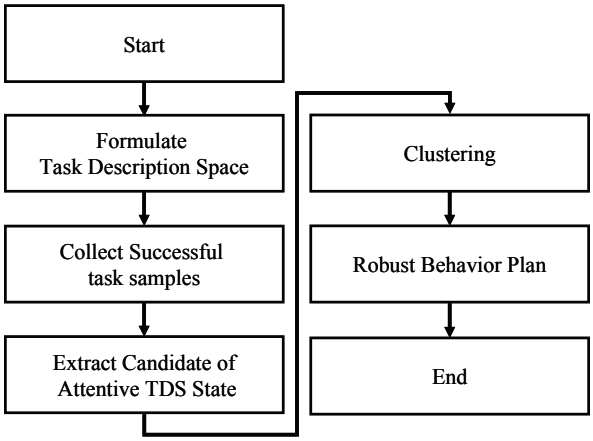


그림 1. 강인 행동 계획 생성 순서

2. 강인 행동 계획의 생성

강인 행동 계획은 로봇이 동적인 주변 환경과 불완전한 인지 능력, 그리고 불완전한 행동 출력 능력 하에서도 강인한 태스크 수행을 위한 방법을 제공한다. 강인 행동 계획은 태스크 성취를 위한 명의상 경로를 상태-행동 쌍의 순서로 갖는다. 태스크 수행 경로를 자동으로 찾기 위해서 다양한 형태의 성공적인 태스크 에피소드와 경로를 포함하고 있는 태스크 에피소드 라이브러리로부터 성공적으로 태스크를 수행한 상태-행동 쌍들 중에서 주목할만한 상태-행동의 쌍을 추상화하고, 명의상 태스크의 행동 순서를 작성한다. 태스크를 수행하는 동안 명의상 순서는 작업 환경[13]의 현재 상황에 따라 합리적으로 재배치 된다. 그림 1은 강인 행동 계획을 생성하는 순서이다. 보편성을 유지하면서 이동 로봇의 'Box-Pushing-into-a-Goal'의 태스크를 사용해 강인 행동 계획 알고리즘의 각 단계에 대해 설명하도록 하겠다.

2.1 Box-Pushing-into-a-Goal (BPIG)

강인 행동 계획 방법을 설명하기 위해 본 논문은 로봇이 방을 가로질러 상자를 미는 태스크를 선택하였다. 이것은 로봇이 한 위치에서 다른 위치로 상자를 운반하는 단순한 태스크이다. 서비스 로봇(이동 로봇)이 Box-Pushing(BP) 태스크를 수행하기 위해서는 몇 가지 어려움이 존재한다. 첫째, 이동 로봇이 상자를 미는 동안 상자가 예상치 못하게 밀리거나 회전하는 상황이 발생한다. 둘째, 로봇은 상자, 목표, 자기 자신의 공간적인 위치 관계를 정의하는 과정이 필요하다. 셋째, 로봇은 자신의 현재 상황을 놓친 경우 다시 그 상황을 찾아내는 과정이 필요하다.

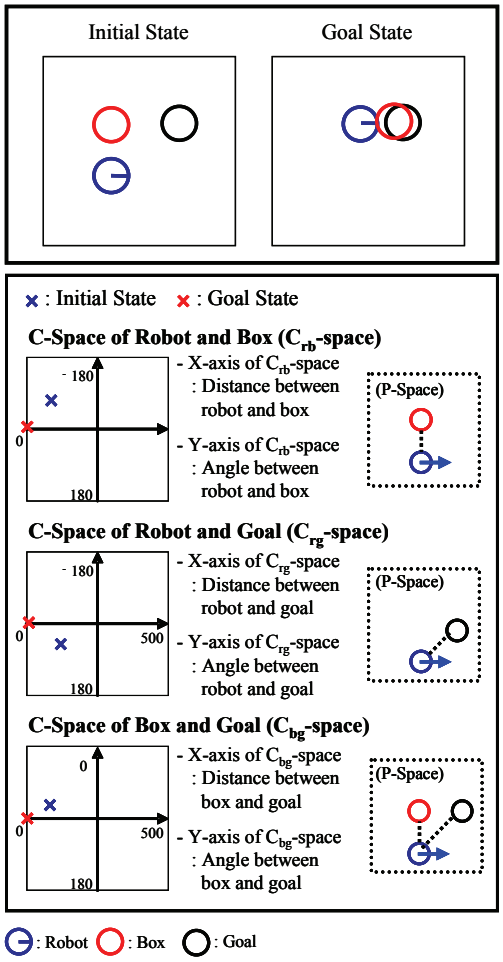


그림 2. BPIG 태스크에 대한 태스크 설명 공간 생성

BP 태스크는 BPIG 태스크와는 달리 목표의 위치와 관련해 상자와 로봇 자신의 공간적인 관계를 찾는 것이 불필요하다. 단순한 BP 태스크의 경우 조차 학습과 계획은 하나의 통일된 학습 구조로 성공하는 것이 어렵다. 이 때, BP 태스크는 실제 몇 개의 하위 태스크와 하위 목표로 구분될 수 있다. 때문에 로봇이 각각의 하위 태스크를 구분해서 학습하는 경우 실패했던 많은 문제를 성공적으로 수행할 수 있다. 이 접근은 개발 과정[3]에 대한 입력의 복잡성을 낮추는데 적합하다. 동일한 태스크에 대해 여러 가지 상황을 증명하고 하위 태스크들이 갖고 있는 특징을 구별하여 중심이 되는 하위 태스크 또는 하위 목표를 정확히 취득하는 것이 태스크 학습과 계획의 요점이다. 하지만 동일한 태스크에 대해 여러 가지 상황을 증명할 수 있는 각각의 하위 태스크를 구별하는 것은 쉬운 작업이 아니다.

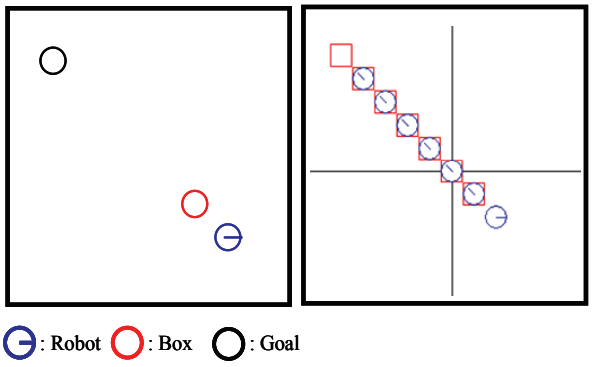


그림 3. BPIG 태스크에 A* 알고리즘을 적용한 에피소드

2.2 강인 행동 계획 알고리즘

2.2.1 태스크 설명 공간과 성공한 에피소드의 수집

태스크 설명 공간(Task Description Space - TDS)은 강인 행동 계획 생성기가 이해할 수 있도록 태스크를 설명하는 공간이다. 태스크 설명 공간은 물리 공간(physical space - P_{space})과 상태 공간(configuration space - C_{space})으로 이루어져 있다. 상태 공간에서 행동할 순서를 찾고 물리 공간에서 그 비용을 계산한다. 일반적으로 로봇 분야에서 말하는 상태 공간의 기본 벡터는 서로 독립적이다[2][15]. 그러나, 여기서 사용한 태스크 설명 공간에 대한 상태 공간의 기본 벡터들은 비독립적이다. 이것은 태스크 설명 공간에서 상태 공간의 상태 벡터가 물리 공간의 상태 벡터와 일대일로 부합되지 않는다는 것을 의미한다. 다시 말해, 태스크 설명 공간에 대한 상태 공간의 상태 벡터는 물리적으로 의미를 갖지 못하기 때문에 상태 공간의 상태 벡터가 물리적으로 어떤 의미를 갖는지 확인하는 작업이 필요하다. 이 작업은 물리 공간에서 확인하는 작업을 통해 수행한다. 상태 공간의 단일 상태는 물리 공간의 기준 좌표에 따라 다른 특징을 갖는다. 이것은 물리공간에서 서로 다른 위치에 놓인 두 에피소드가 태스크 수행의 관점에서 동일한 의미를 갖는다면 상태 공간에서는 동일한 위치로 표현된다. 상태 공간은 강인 행동 계획이 요구하는 성공적인 태스크 에피소드(또는 경로)를 강인 행동 계획 생성기가 쉽게 취득할 수 있도록 도와준다. 강인 행동 계획기에 필요한 다양한 에피소드의 성공 예를 수집하기 위해서 우리는 상태 공간에서 각 에피소드의 성공 경로를 A*와 같은 경로 계획기를 이용하여 초기 상태부터 목표 상태까지의 최소 경로를 찾도록 한다.

BPIG 태스크의 상태 공간(C_{space})은 6차원 공간이다.

C_{space}는 그림 2와 같이 d_{rb}, a_{rb}, d_{rg}, a_{rg}, d_{bg}, a_{bg}로 정의된다. 여기서 d는 거리를 나타내고, a는 각도를 나타낸다. BPIG 태스크에 대한 상태 공간 $\mathbf{S}$ 는

$$\mathbf{S} = [d_{rb}, a_{rb}, d_{rg}, a_{rg}, d_{bg}, a_{bg}] \quad (1)$$

와 같이 정의된다. BPIG 태스크 상태 공간에 대한 $\mathbf{S}$ 의 요소들은 서로 비독립적일 수 있다. 에피소드의 초기상태($\mathbf{S}_{init}$)에서 목표 상태($\mathbf{S}_{goal}$)까지 태스크 설명 공간의 경로는 BPIG 태스크 상태 공간에서 선택된 노드(node)들에 A^* 알고리즘을 적용해 찾아낸다. 그리고, 물리 공간에서 물리적으로 가능한 노드들의 비용을 계산한다. 그림 3은 A^* 알고리즘을 이용해 BPIG 태스크의 성공적인 경로를 찾아낸 결과이다.

2.2.2 태스크 설명 공간의 관심 있는 상태 추출

태스크 설명 공간에서 성공적으로 태스크를 수행한 에피소드의 모든 노드들은 태스크 설명 공간의 관심 있는 상태가 될 수 있는 후보가 된다. BPIG 태스크에서 상태 공간은 동일한 13,500개의 하위 공간으로 나뉜다. 본 논문에서 수행한 13,500개의 실험에 대한 초기 상태는 13,500개의 하위 공간으로부터 추출하였다. 이 때, 태스크 설명 공간에 A^* 알고리즘을 적용해 1,512개의 성공적인 태스크 에피소드(또는 경로)를 찾았고 1,512개의 태스크 에피소드로부터 13,124개의 태스크 설명 공간 상태 벡터 $\mathbf{S}_1, \ldots, \mathbf{S}_{13,124}$를 생성하였다. 여기서 $\mathbb{S}_{init}$ 은 13,124개의 태스크 설명 공간 상태 벡터를 포함하는 집합이다. $\mathbb{S}_{init}$ 에 있는 상태 벡터들은 여러 가지로 구별이 가능하다. 예를 들어, $\mathbf{S}_{goal}$ 은 성공적으로 태스크를 수행한 수많은 태스크 에피소드들에 존재한다. 반면, 한 번만 발생하는 상태도 존재한다. 강인 행동 계획의 목적은 여러 번 반복해서 발생하는 상태를 이용해 일반적인 계획을 추출하는 것이다. 이를 위해 $frequency(\mathbf{S}_i, \square)$ 을

$$frequency(\mathbf{S}_i, \square) \equiv number\ of\ \mathbf{S}_i \ satisfying\ (\mathbf{S}=\mathbf{S}_i) \wedge (\mathbf{S} \in \square) \quad (2)$$

와 같이 정의했다. 그리고 $\mathbb{S}_1$ 와 $\mathbb{S}$ 는

$$\mathbb{S}_1 = \{\mathbf{S}_i \mid (\mathbf{S}_i \in \mathbb{S}_{init}) \wedge (frequency(\mathbf{S}_i, \mathbb{S}_{init}) \geq 2)\}, \quad (3)$$

과

$$\mathbb{S} = \{S_i \mid (S_i, S_j \in (S)_1) \wedge (S_i \neq S_j).\} \quad (4)$$

으로 정의하였다.

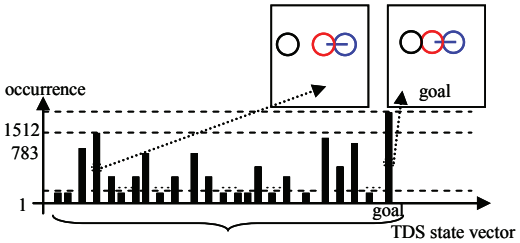


그림 4. BPIG 태스크에서 $\mathbb{S}_{init}$의 TDS 상태 히스토그램

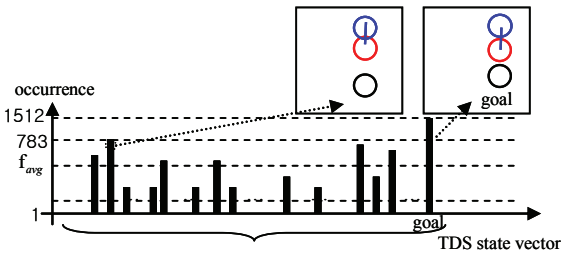


그림 5. BPIG 태스크에서 $\mathbb{S}_1$ 의 TDS 상태 히스토그램

식 (3)과 (4)에서 $\mathbf{S}_i$ 와 $\mathbb{S}$ 는 빈도수가 2보다 큰 상태 집합에만 적용된다. 그리고 이 집합은 중복된 상태를 허용하지 않는다. 발생 빈도의 평균값 f_{th} 는

$$f_{th} = \frac{|\mathbb{S}_1|}{|\mathbb{S}|}, \quad (5)$$

와 같이 계산된다. 이 때, 집합 $|\mathbb{S}|$은 $\mathbb{S}$ 의 크기(cardinality)이다. 강인 행동 계획은 f_{th} 보다 큰 빈도수를 가지고 있는 상태들로부터 만들어지고, 그런 후보 상태들의 집합 $\mathbb{S}_{interesting}$ 은

$$\mathbb{S}_{interesting} = \{S_i \mid (S_i \in \mathbb{S}) \wedge (frequency(S_i, \mathbb{S}) \geq f_{th})\} \quad (6)$$

이다. 다시 말해, $\mathbb{S}_{interesting}$은 발생 빈도로부터 만들어진다. 그 이유는 다양한 초기 상태로부터 동일한 목표 상태로 태스크를 진행하는 동안 로봇은 동일하거나 유사한 상황이 다수 발생하기 때문에 빈번하게 발생하는 상황을 목표 성취를 위한 하위 목표로 볼 수 있기 때문이다. 그림 4는 BPIG 태스크에서 567개의 다른 태스크 설명 공간 벡터를 포함하는 $\mathbb{S}_{init}$의 태스크 설명 공간 상태 벡터들의 돗수 분포를 보여준다. 그림 5는 식(3) 에 있는 $\mathbb{S}_1$의 서로 다른 상태 벡터 418개의 돗수 분포를 보여준다. 그리고 BPIG 태스크에서 식(6)을 만족하는 $\mathbb{S}_{interesting}$은

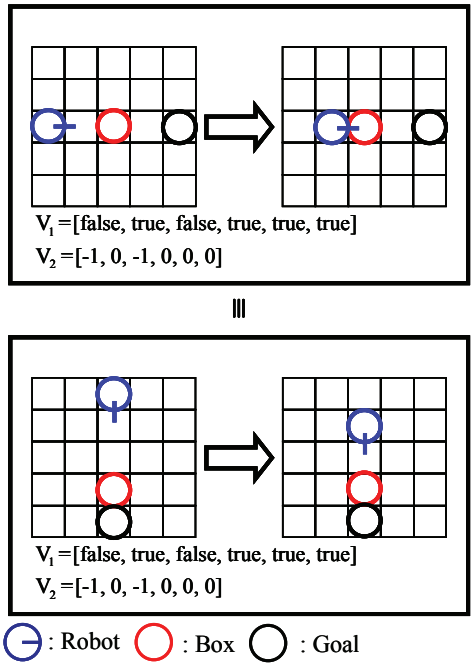


그림 6. BPIG 태스크에서 $F_{SAS'}$ 의 생성 예

61개가 존재한다. 각 상태 벡터 $S' \in \mathbb{S}_{interesting}$ 에는 몇 가지 행동을 통해 S'이 될 수 있는 이전 상태 벡터의 집합 $\mathbb{S}'_{transit}$ 이 존재한다. BPIG 태스크에서 $\mathbb{S}'_{transit}$ 의 크기는 $\sum_{S' \in \mathbb{S}_{interesting}} |\mathbb{S}'_{transit}| = 8,593$ 와 같다.

2.2.3 상태 전위 튜플과 클러스터링

상태 전위 튜플(state transition tuple) $T_{SAS'}$ 는

$$T_{SAS'} \equiv (S, A, S') \tag{7}$$

으로 정의된다. 이 때, A는 S 인 상태에서 S' 인 상태로 가기 위한 행동을 말한다. 상태 전위 튜플 $T_{SAS'}$ 으로부터 상태 벡터의 변화 여부, 상태 벡터 변화 요소, 변화된 값의 증감 여부와 같은 정보를 추출할 수 있다. $T_{SAS'}$ 의 특징 벡터는

$$F_{SAS'} = [V_1, V_2] \tag{8}$$

$$V_1 = [f_1^1, \ldots, f_n^1], \tag{9}$$

$$V_2 = [f_1^2, \ldots, f_n^2], \tag{10}$$

으로 정의된다. 이 때, for all $i = 1$ to n, f_i^1 and f_i^2 은

$$f_i^1 = \begin{cases} true, & \text{if } S_i = S'_i, \\ false, & \text{otherwise,} \end{cases}$$

와

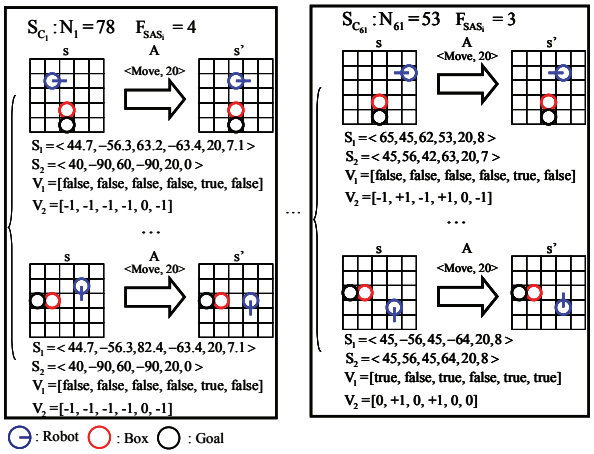


그림 7. 다른 $T_{SAS'}$이 같은 $F_{SAS'}$을 갖는 예

$$f_i^2 = \begin{cases} 1, & \text{if } S_i < S'_i \\ 0, & \text{if } S_i = S'_i \\ -1, & \text{if } S_i > S'_i \end{cases}$$

으로 정의된다. 여기서 S_i와 S'_i는 $T_{SAS'} = (S, A, S')$의 i번째 S와 S' 을 나타낸다. BPIG 에피소드에서는 8,593개의 $T_{SAS'}$와 $F_{SAS'}$의 쌍이 발견되었다. 그림 6은 $F_{SAS'}$의 예를 보여준다. 그리고 다른 T_{SAS}은 그림 7에서 보여지는 것처럼 같은 $F_{SAS'}$이 될 수 있다. 그러므로 상태 전위 튜플에는 같은 $F_{SAS'}$을 갖는 $T_{SAS'}$이 동일한 클러스터에 포함될 수 있다. 본 논문에서는 이와 같은 방법을 통해 클러스터 생성을 공식화하였다. BPIG 태스크는 61개의 그룹과 8,593개의 $T_{SAS'}$가 생성되었다. 그리고, 식(8)에 의 해 14개의 클러스터로 분류되었다.

2.2.4 일반화 (Generalization)

식(8)에 의해 분류된 클러스터들로부터 일반화하는 과정을 수행한다. 본 논문에서는 [6]에서 언급한 concept learning의 성공 예제에 대한 general-to-specific ordering 방법으로 표 1과 같이 수행하였다. 다시 말해, 성공

표 1. FISD-S ALGORITHM

FIND-S ALGORITHM

1. Initialize to h the most specific hypothesis in H
2. For each positive training instance x
   - For each attribute constraint a_i in h
     - If the constraint a_i is satisfied by x
     - Then do nothing
     - Else replace a_i in h by the next more general Constraint that is satisfied by x
3. Output hypothesis h

표 2. BPIG 태스크의 주목할 만한 TDS 상태 벡터

14 ATTENTIVE TDS STATE VECTORS FOR BPIG TASK

Cluster(Ω_i)	Attentive TDS State Vector(S^{Ω_i})
Ω_1	< 20, 0, 20, 0, 0, 0 >
Ω_2	< 20 , 0, x, 0, x, 0 >
Ω_3	< x, 0, x, 0, x, 0 >
Ω_4	< 20, x, x, x, x, 0 >
Ω_5	< 20, x, x, x, x, 0 >
Ω_6	< 20, x, 63, x, 40, 71 >
Ω_7	< 20, x, 44, x, 60, 63 >
Ω_8	< 20, x, 82, x, 80, 76 >
Ω_9	< 20, x, 28, x, 20, 45 >
Ω_{10}	< 28, x, 44, x, 20, 18 >
Ω_{11}	< 28, x, x, x, x, x >
Ω_{12}	< 44, x, 121, x, 80, 17 >
Ω_{13}	< 44, x, x, x, x, x >
Ω_{14}	< 44, x, 101, x, 60, 15 >

x implies 'don't care'

에피소드를 이용한 more-general-than ordering 방법을 통해 가장 특정한 상태 벡터를 일반화시키는 과정을 수행하였다. 이 알고리즘은 주목할만한 태스크 설명 공간의 상태 벡터 S^{Ω_j} 을 갖도록 한다. 이 때, S^{Ω_j} 은

$$S^{\Omega_i} = [d_{rb}^{\Omega_i}, a_{rb}^{\Omega_i}, d_{rg}^{\Omega_i}, a_{rg}^{\Omega_i}, d_{bg}^{\Omega_i}, a_{bg}^{\Omega_i}]$$

$$d_{rb}^{\Omega_i} = \begin{cases} d_{rb}, & \text{if } d_{rb}' = d_{rb} \\ \times, & \text{otherwise} \end{cases}$$

$$a_{rb}^{\Omega_i} = \begin{cases} a_{rb}, & \text{if } a_{rb}' = a_{rb} \\ \times, & \text{otherwise} \end{cases}$$

$$d_{rg}^{\Omega_i} = \begin{cases} d_{rg}, & \text{if } d_{rg}' = d_{rg} \\ \times, & \text{otherwise} \end{cases} \quad (11)$$

$$a_{rg}^{\Omega_i} = \begin{cases} a_{rg}, & \text{if } a_{rg}' = a_{rg} \\ \times, & \text{otherwise} \end{cases}$$

$$d_{bg}^{\Omega_i} = \begin{cases} d_{bg}, & \text{if } d_{bg}' = d_{bg} \\ \times, & \text{otherwise} \end{cases}$$

$$a_{bg}^{\Omega_i} = \begin{cases} a_{bg}, & \text{if } a_{bg}' = a_{bg} \\ \times, & \text{otherwise} \end{cases}$$

와 같이 표현된다. 표 2는 BPIG 태스크의 클러스터 Ω_i ,i=1,2,...,14 에 대한 주목할만한 태스크 설명 공간의 상태 벡터(S^{Ω_i})를 보여준다. S^{Ω_i} 은 다음과 같이 결합과 분리를 수행할 수 있다.

결합: $S^{\Omega_i} = S^{\Omega_j}$이면 클러스터 Ω_i 와 Ω_j 을 결합한다.

분리: 몇 개의 Ω_i 와 Ω_j에 대해 S^{Ω_j}이 S^{Ω_i}보다 더 일반적이고, Ω_i의 특징 벡터가 Ω_j 의 특징 벡터와 같으면 그에 해당하는 $T_{SAS'}$ 와 Ω_i의 F_{SAS}은 Ω_i으로부터 분리되어 Ω_j에 합쳐 진다. 그림 8과 그림 9는 BPIG 태스크에 대한 결합과 분리의 예를 보여준다. 결합과 분리의 과정을 거치고 난 후 그림 10에서 보이는 것처럼 12개의 클러스터만 존재하게 된다.

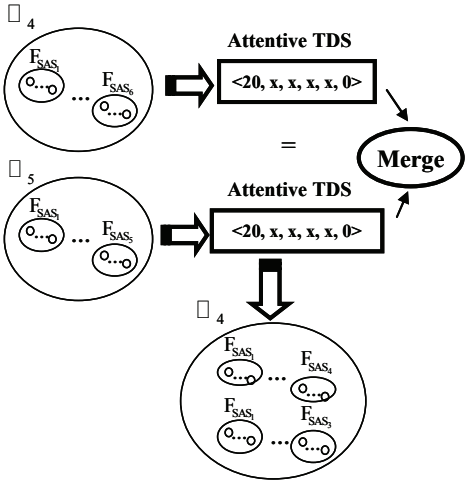


그림 8. 다른 Ω_4 이 같은 Ω_5 을 갖는 예

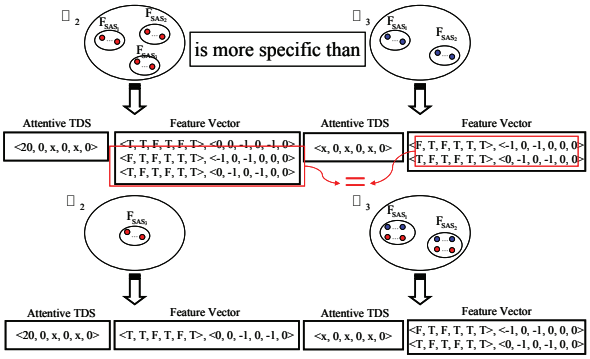


그림 9. Ω_2에서 분리되어 Ω_3에 결합되는 예

2.2.5 행동 계획의 트리 구조 순서

결합과 분리를 통해 남은 모든 클러스터들은 순서를 갖는 트리 구조의 행동 계획 단위로 만들 수 있다. 또한

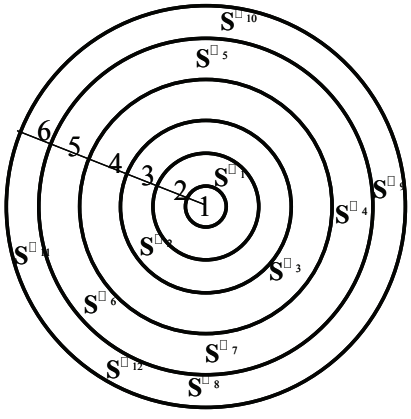


그림 10. 다른 Ω_4이 같은 Ω_5 을 갖는 예

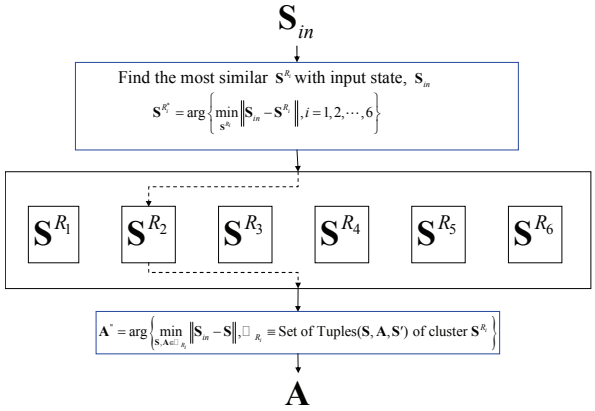


그림 11. BPIG의 강인 행동 계획 트리 구조

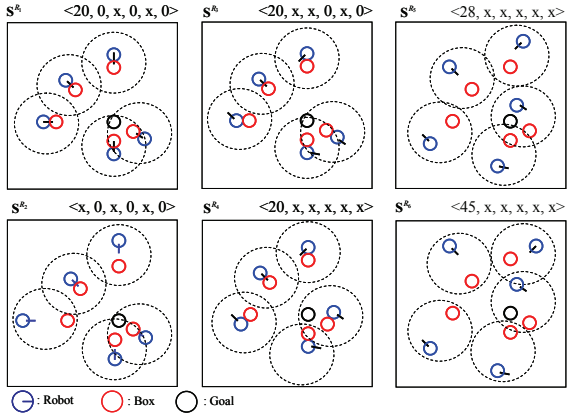


그림 12. BPIG의 강인 행동 계획

표 3 시뮬레이션 환경

THE ENVIRONMENT FOR SIMULATION

Robot	Simulation Environment		Real Environment	
로봇의 크기	Shape	Size(pixel)	Shape	Size(cm)
Working Space	Rectangle	300 X 300	Rectangle	500 X 500
Robot	Circle	10 (R)	Rectangular parallelepiped	25(W) X 40(D) X 30(H)
Box	Circle	10 (R)	Cylinder	10(R) X 15(H)
Goal area	Circle	10 (R)	Rectangle	45(W) X 45(W)

이 클러스터들을 주목할만한 태스크 상태 공간의 상태 벡터와 목표 벡터 사이의 거리 차이로 더욱 결합하여 트리 구조의 행동 계획 단위로 만드는 것도 가능하다. BPIG 태스크에서는 그림 10에 있는 12개의 클러스터들을 목표와의 거리 차이로 클러스터링 하면 6개의 클러스터가 만들어진다. 모든 상태 전위 튜플은 그에 해당하는 행동과 주목할 만한 태스크 상태 공간에 대한 상태 벡터를 포함하고 있다. 때문에 그림 11에서 행동 $\mathbf{A}^*$ 은

$$\mathbf{A}^* = \arg\{\min_{\mathbf{S},\mathbf{A}\in\square_{R_i}} \|\mathbf{S}_{in}-\mathbf{S}\|, \square_{R_i} \equiv \text{Set of tuples}(\mathbf{S},\mathbf{A},\mathbf{S}') \text{ of cluster } \mathbf{S}^{R_i}\} \quad (12)$$

과 같은 과정을 통해 선택된다.

3. 시뮬레이션과 실험

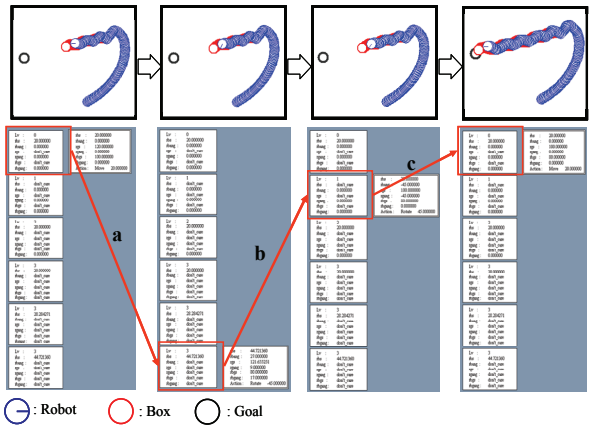
Box-Pushing-into-a-Goal 태스크에 대한 강인 행동 계획은 그림 12와 같이 6개의 하위 목표를 얻을 수 있다.

각 강인 행동 계획은 말로 표현하면 다음과 같이 해석할 수 있다.

Rule 1 ($\mathbf{S}^{R_1}$): 로봇이 상자에 접해있고 방향이 상자와 목표에 일직선이 되는 방향으로 놓여 있으면 로봇은 행동 $\mathbf{A}^1$을 수행한다. $\mathbf{A}^1$은 식(12)로부터 얻을 수 있다. ($\mathbf{A}^1$ 은 "앞으로 밀기"를 의미한다.)

Rule 2 ($\mathbf{S}^{R_2}$): 로봇이 상자와 접해있거나 또는 로봇의 방향이 상자와 목표에 일직선이 되는 방향으로 놓여 있으면 로봇은 행동 $\mathbf{A}^2$을 수행한다. ($\mathbf{A}^2$은 "밀기 또는 앞으로 이동"을 의미한다.)

Rule 3 ($\mathbf{S}^{R_3}$): 로봇이 상자에 접해있고 로봇-상자-목표가 일직선상에 있지만 로봇의 방향이 상자와 목표에 일직선이 되는 방향이 아닌 경우, 로봇은 $\mathbf{A}^3$을 수행한다. ($\mathbf{A}^3$은 일직선이 되도록 "회전"하는 것을 의미한다.)



a. 상자를 미는 동중 상자가 밀리면 다시 상자 가까이 이동
b. 상자 가까이 이동한 후 다시 상자와 마주치도록 회전
c. 회전 후 다시 상자를 밈

그림 13. BPIG 태스크에 대한 시뮬레이션 결과

표 4. 강인 행동 계획과 A*의 강인성 비교 실험

강인 행동 계획과 A* 알고리즘의 강인성 비교 실험 결과

실험 번호	A* Algorithm		실험 번호	Robust Behavior Plan	
	센서 오차 없음	센서 오차 (Gaussian $N(0,1)$)		센서 오차 없음	센서 오차 (Gaussian $N(0,1)$)
1	180	X	1	302	308
2	380	X	2	654	672
3	340	X	3	622	630
4	300	X	4	408	436
5	240	X	5	384	408
6	320	X	6	602	664
7	340	X	7	636	642
8	220	X	8	184	192
9	320	X	9	396	420
10	100	X	10	310	216
11	220	X	11	734	796
12	120	X	12	198	182
13	200	X	13	548	560
14	120	X	14	222	240
15	180	X	15	264	272
16	200	X	16	844	854
17	220	X	17	214	222
18	100	X	18	298	288
19	180	X	19	300	318
20	220	X	20	970	964

※ 'x'는 태스크 수행 실패를 나타낸다.

Rule 4 (S^{R_4}): 로봇이 상자에 접해있지만 로봇-상자-목표가 일직선상에 없고 로봇의 방향이 상자와 목표에 일 직선이 되는 방향이 아닌 경우, 로봇은 행동 A^4을 수행한다. (A^4은 로봇-상자-목표가 일직선이 되도록 "상자 주위로 이동"을 의미한다.)

Rule 5 (S^{R_5}): 로봇이 상자에서 멀리 떨어져 있고 로봇-상자-목표가 일직선상에 없고 로봇의 방향이 상자와 목표에 일직선이 되는 방향이 아닌 경우, 로봇은 행동 A^5을 수행한다. (A^5은 로봇이 상자에 접하도록 상자를 향해 "평범하게 앞으로 이동"을 의미한다.)

Rule 6 (S^{R_6}): 로봇이 상자로부터 더욱 멀리 떨어져 있고 로봇-상자-목표가 일직선상에 없고 로봇의 방향이 상자와 목표에 일직선이 되는 방향이 아닌 경우, 로봇은 행동 A^6을 수행한다. (A^6은 로봇이 상자에 접하도록 상자를 향해 "빠르게 앞으로 이동"을 의미한다.)

BPIG 태스크에 대한 강인 행동 계획을 생성하기 위해 표 3과 같은 물리 환경에서 시뮬레이션을 수행하였다. 그림 13은 강인 행동 계획이 성공적으로 동작하는 예를 보여준다. 또한, 본 논문의 실험은 센서의 불확실성이 존재하는 환경에서 수행하였다. 여기서 센서의 불확실성은 로봇의 이동에 대한 표준 정규 분포 $N(0,1)$을 갖는 누적 오차를 말한다. 로봇이 이동하는 동안 발생하는 오차가 누적되었을 때, 로봇의 태스크 성공 여부를 판단하였다. 표 4는 센서와 행동의 불확실성이 존재하는 환경에서 A* 알고리즘이 실패하는 것에 반해 강인 행동 계획 알고리즘은 임의의 초기 상태에서 BPIG 태스크를 완벽하게 수행하는 것을 보여준다. 또한, 본 논문에서는 그림 14와 같이 직접 제작한 이동 로봇인 CUBO에 시뮬레이션을 통해 생성한 강인 행동 계획 알고리즘을 탑재하여 실험하였다. CUBO는 동물학 행동 기반 구조(Ethology-based Action Selection mechanism –EASE)[14]로 제어된다. 시뮬레이션과 실험에 대한 비디오 클립은 웹사이트(http://isquare.ha nyang.ac.kr/rbp/vclip.html)에서 확인할 수 있다. 그리고, 또 다른 태스크인 "Obstacle-Avoidance-while-using-Box (OAWPB)"에 강인 행동 계획을 적용한 비디오 클립도 함께 확인할 수 있다. OAWPB에 대한 자세한 내용은 생략하기로 하겠다.

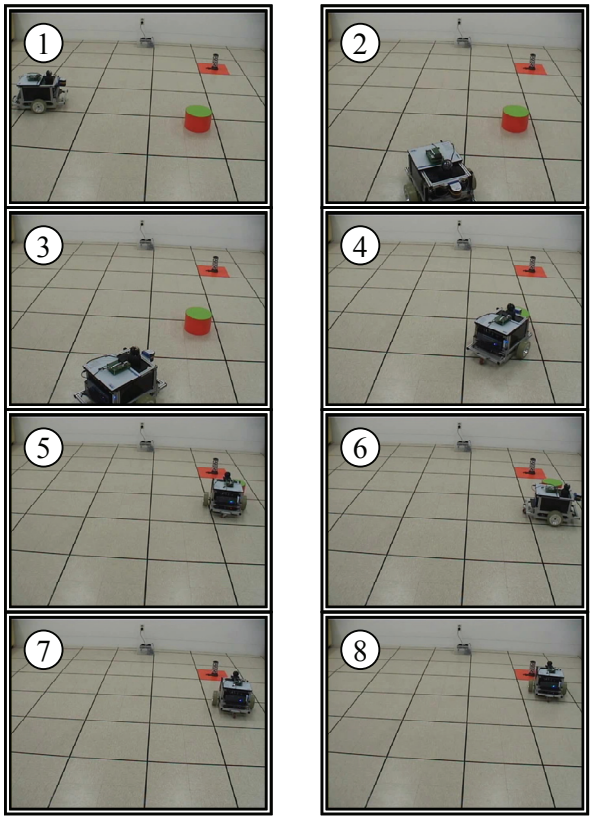


그림 14. BPIG에 대한 실제 로봇 실험

4. 결과 및 향후 계획

성공한 에피소드 라이브러리로부터 반응 계획을 자동으로 생성하는 강인 행동 계획을 제안했다. 강인 행

동 계획기는 에피소드 라이브러리를 RRT 또는 A*와 같은 경로 계획기를 이용하여 생성한다. 그리고 에피소드 라이브러리로부터 주목할 만한 상태들을 추출한다. 강인 행동 계획은 추출한 상태들에 포함된 상태-행동 집합으로 만들어진다. 상태-행동 집합을 순서가 있는 트리 구조로 만들고 현재 상태와 가장 가까운 이웃 상태로부터 행동을 선택해 목표를 완수한다. 따라서 강인 행동 계획은 예외 상황에서도 로봇이 태스크를 성취하도록 도와준다. "Box-Pushing-into-a-Goal" 태스크와 "Obstacle-Avoidance-while-Pushing-Box" 태스크에 대한 강인 행동 계획을 생성하고 직접 제작한 이동 로봇에 적용하여 실험하였다. 두 실험 모두 시뮬레이션을 통해 생성한 강인 행동 계획이 노이즈(noise)가 존재하는 실제 환경에서도 성공적으로 태스크를 수행하는 것을 확인하였다.

앞으로 실제 로봇의 다양한 태스크에 적용하는 실험을 진행하면서 강인 행동 계획 방법을 개선시킬 계획이다. 개선 방법으로 첫째, 적은 횟수의 에피소드로 강인 행동 계획을 생성한다. 둘째, 성공한 에피소드뿐만 아니라 실패한 에피소드를 이용해 강인 행동 계획을 생성한다. 셋째, 점진적으로 에피소드 라이브러리를 증가시켜 향상된 강인 행동 계획을 만들어 가는 학습 방법 등이 있다.

참고문헌

[1] J. J. Bryson, "Intelligence by design:principles of modularity and coordination for complex adaptive agents," Ph.D. dissertation, Massachusetts Institute of Technology, 2001.

[2] H. Choset, K. Lynch, S. Hutchinson, G.Kantor, W. Burgard, L. Kavraki, and S. Thrun, Principles of Robot Motion: Theory, Algorithms, and Implementaions. MIT Press, 2005.

[3] J. H. Connel and S. Mahadevan, ROBOT LEARNING. Kluwer Academic, June 1993.

[4] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Transactions on Robotics and Automation, vol. 12, no. 8, pp. 566-580, 1996.

[5] S. M. Lavalle and J. J. Kuffner, " Randomized Kinodynamic planning," The International Journal of Robotics Research, vol. 20, no. 5, pp. 378-400, 2001.

[6] T. M. Mithcel, Machine Learning. McGraw-Hill, 1997.

[7] N. J. Nilsson, Artificial Intelligence: A New Synthesis. Morgan Kaufmann, 1998.

[8] R. Pfeifer and C. Scheier, Understanding Intelligence. MIT Press, 2001.

[9] B. Prasad, "A planning system for blocks-world domain," in ACS/IEEE International Conference on Computer Systems and Applications. IEEE Computer Society, pp. 59-64, 2001.

[10] S. Russell and P. Norving, Artificial Intelligence: A Modern Approach. Prentice Hall, 2003.

[11] L. Steels, Embodied Artificial Intelligence. Springer Berlin/Heidelberg, 2004, vol. 3139, ch. The Autotelic Principle, pp. 231-242.

[12] M. Stolle and G. G. Atkeson, "Policies based on trajectory libraries," in IEEE International Conference on Robotics and Automation, Orlando, Florida, USA, pp. 3344-3349, 2006.

[13] I. H. Suh, M. J. Kim, S. Lee, and B. –J. Yi, "A novel dynamic priority-based action-selection-mechanism integrating a reinforcement learning," in Proceedings of IEEE Conference on Robotics and Automation, pp. 2639-2646, 2004.

[14] I. H. Suh, S. Lee, W. Y. Kwon, and Y. J. Cho, "Learning of action patterns and reactive behavior plans via a novel two-layered ethology-based action selection mechanism," in IEEE/RSJ International Conference on Intelligence Robots and Systems, 2005.

[15] S. Thrun, W. Burgard, and D. Fox, Probabilistic Robotics. MIT Press, 2005.

이 상 형

2005 한양대학교 전자 및 컴퓨터공학과(공학사)

2005~현재 한양대학교 정보통신학과 석사과정

관심분야: 지능응용시스템, 인공지능

이 상 훈

1994 한양대학교 수학과 (이학사)
1997 한양대학교 전자계산학 전공(공학석사)
2006 한양대학교 전자, 전기, 제어계측과(공학박사)
관심분야: 로봇 지능, 행동선택방법, 로봇 S/W 프레임웍, 지능응용시스템

서 일 홍

1977 서울대학교 전자공학과 (공학사)
1979 한국과학기술원 전기 및 전자공학(공학석사)
1982 한국과학기술원 전기 및 전자공학(공학박사)
1985 대우중공업 기술연구소
1985~현재 한양대학교 교수
관심분야: 지능응용시스템, 인공지능, 로봇공학