

복수 무인기를 위한 POMDP 기반 동적 임무 할당 및 정찰 임무 최적화 기법

(A POMDP Framework for Dynamic Task Allocation and Reconnaissance of Multiple Unmanned Aerial Vehicles)

김 동 호 [†] 이 재 송 ^{**} 최 재 득 ^{**} 김 기 응 ^{***}
(Dongho Kim) (Jaesong Lee) (Jaedeug Choi) (Kee-Eung Kim)

요 약 최근 무인항공기의 제작 기술이 발전함에 따라, 농업, 재해 관측용 등의 민간 용도 뿐만 아니라 정찰 및 공격 등의 군사적 목적으로 다수의 무인기를 사용하는 다양한 시도가 진행되고 있다. 그러나 다수의 무인기를 사용할 때에 각 무인기를 사람이 직접 제어하는 데에는 어려움이 많으므로, 주어진 목표를 달성하기 위해서 자율적으로 협력하며 효과적인 행동을 수행하는 알고리즘의 개발이 필수적이다. 이러한 문제는 순차적 의사결정 문제로 생각할 수 있으며, 마코프 의사결정 과정(Markov Decision Processes; MDPs)과 이를 부분적 혹은 부정확한 관찰값을 다룰 수 있도록 확장한 부분관찰 마코프 의사결정 과정(Partially Observable MDPs; POMDPs) 등의 대표적인 의사결정이론 모델을 이용하여 복잡하고 불확실한 환경에서의 의사결정 문제를 통계적으로 다룰 수 있다. 본 논문에서는 복수의 무인기를 이용할 때 동적 임무 할당 및 정찰 임무 문제를 POMDP를 이용하여 효율적으로 최적화할 수 있음을 보이고, 센서의 관찰값에 오차가 발생할 수 있는 경우, MDP에 비해 POMDP를 이용할 때 더 좋은 성능을 얻을 수 있음을 보인다. 또한 실제 쿼드콥터(quadcopter)를 이용하여 POMDP 정책이 실제 환경에서도 잘 동작함을 시뮬레이션을 통해 입증하였다.

키워드 : 무인기, 순차적 의사결정, POMDP

Abstract Interest in unmanned aerial vehicles (UAVs) for military use such as surveillance and target tracking or civil applications such as agriculture and firefighting has increased in recent years, due to significant progress in their development. Since there are still required human operators to remote-control multiple UAVs, autonomous control of UAVs has been significantly studied. Autonomous planning and control of multiple UAVs can be viewed as a sequential decision problem, which can be modeled as Markov decision processes (MDPs) or partially observable Markov decision processes (POMDPs) which extend MDPs by allowing partial or uncertain observations. This paper discusses the application of POMDPs to optimize the control of UAVs in reconnaissance flights with dynamic task allocation, and we experimentally show that the policies computed from POMDPs perform better than those from MDPs under uncertain observations. The applicability of the POMDP framework is also demonstrated by the realistic simulation using multiple quadcopters.

Key words : unmanned aerial vehicles, sequential decision making, POMDP

· 본 연구는 국방과학연구소 복수 무인기의 분산시스템 연구(09-01-03-04)의 연구비 지원으로 수행하였습니다.

· 이 논문은 정보과학회지 제29권 제2호(2011년 2월호)에 '부분관찰 마코프 의사결정 과정을 이용한 지능형 에이전트 구현'의 제목으로 게재된 원고를 확장한 것임

[†] 정 회 원 : University of Cambridge 박사후연구원
dkim@ai.kaist.ac.kr

^{**} 학생회원 : KAIST 전산학과
jaesong@ai.kaist.ac.kr
jdchoi@ai.kaist.ac.kr

^{***} 종신회원 : KAIST 전산학과 교수
kekim@cs.kaist.ac.kr
(Corresponding author임)

논문접수 : 2011년 12월 12일

심사완료 : 2012년 2월 16일

Copyright©2012 한국정보과학회 : 개인 목적이나 교육 목적의 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지 : 소프트웨어 및 응용 제39권 제6호(2012.6)

1. 서론

무인기(無人機; unmanned aerial vehicles; UAVs)는 사람이 탑승하지 않는 비행체를 말한다. 무인기는 지상에서의 로봇에 비해 기동력이 높고 관측 범위가 넓은 장점으로 인해 농업, 재해 관측용의 민간 용도 뿐만 아니라, 정찰 및 공격 등의 군사적 목적으로 큰 주목을 받고 있다[1].

기존의 무인기는 일반적으로 고가, 고성능의 정찰 장비를 탑재한 기체를 단독으로 이용하였기에 지상에 원격 조종을 수행하는 조종사가 있는 경우가 많았으나, 복잡한 임무를 효과적으로 수행하기 위해 여러 대의 무인기를 동시에 활용해야 하는 필요가 증대되고, 최근에는 무인기에 탑재될 수 있는 저가의 다양한 센서들이 개발됨에 따라, 자율적으로 행동하는 무인기에 대한 새로운 요구가 커지고 있다. 특히 최근에는 전산학 및 인공지능 분야에서 의사결정 기법의 기술 발전에 힘입어 목적을 달성하기 위해 자율적으로 행동할 수 있는 무인기를 개발하기 위한 다양한 시도가 진행되고 있다[2,3]. 즉, 무인기 및 환경의 상태를 파악하기 위해 각각의 무인기에 항법센서, 고도계, GPS, 카메라 등 다양한 센서들이 탑재되고, 이러한 센서들로부터 얻은 정보를 바탕으로 목표를 최대한 효율적으로 달성하기 위한 협동 및 임무 할당, 최적 비행경로의 계획, 비행 중 방해물 및 위험 회피 등 다양한 요구조건을 만족시키도록 자율적으로 행동할 수 있는 알고리즘의 개발이 필수적이다. 이러한 문제는 계속적으로 상태가 변하는 환경 하에서 순차적인 의사 결정을 최적화하는 순차적 의사결정 문제(sequential decision making)로 생각할 수 있다. 특히 마코프 의사결정 과정(Markov Decision Processes; MDPs)[4]이나 MDP를 부정확한 관찰값을 다룰 수 있도록 확장한 부분관찰 마코프 의사결정 과정(Partially Observable Markov Decision Processes; POMDPs)[5]은 순차적 의사결정 문제를 다루는 대표적인 모델로서, 복잡하고 불확실한 환경에서의 의사결정 문제를 효과적으로 다룰 수 있는 장점을 가진다.

이에 따라, 본 논문에서는 POMDP에 기반한 복수 무인기의 정찰 임무를 위한 최적 행동 계산 기법을 제안한다. 이를 위해 복수 무인기의 정찰 임무를 POMDP 문제로 모델링하고, 그 결과로 얻어진 최적 행동 정책이 MDP를 이용한 결과보다 우수함을 보인다. 또한, POMDP를 이용해 얻은 고수준의 행동 정책을 실제 무인기를 이용한 시뮬레이션에 적용하여, 실제 환경에서도 효과적으로 연구 결과가 이용될 수 있음을 보인다.

본 논문은 다음과 같이 구성된다. 제2장에서는 본 연구와 관련된 기존의 연구들을 소개하고, 제3장에서는 본

연구에서 사용된 순차적 의사결정 모델인 MDP와 POMDP를 소개한다. 제4장에서는 복수 무인기의 정찰 임무 문제를 소개하고, POMDP를 이용한 모델링 방법 및 실험 결과를 보인다. 제5장에서는 POMDP 정책을 이용한 실제 비행 시뮬레이션 결과를 소개하고, 제6장에서 본 논문의 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

여러 대의 무인기를 효율적으로 사용하기 위해서는 각각의 무인기가 수행해야 할 임무를 할당하고, 이러한 임무를 수행하기 위해 움직여야 할 경로를 최적화하는 경로 계획 등의 문제들이 존재한다. 이러한 임무 할당 및 최적 경로 문제는 산업공학 분야에서 차량운행 경로 문제(Vehicle Routing Problem; VRP)[6]로 연구되어 왔다. 이는 여러 대의 차량이 다수의 지역에 물건을 배달하는 문제로, 각각의 차량이 방문할 지역을 결정하는 임무 할당 문제와 그 지역들을 어떠한 경로로 다닐 것인지를 결정하는 최적 경로 문제가 함께 포함되어 있다. 또한 차량의 출발 및 도착지에 대한 제약, 각 지역의 요구를 만족시켜야 하는 제약, 그리고 하나의 차량이 운반할 수 있는 물건의 양에 대한 제약을 만족시키는 동시에 차량의 운행 비용을 최소화하는 것을 목적으로 한다.

이러한 VRP의 최적해를 구하기 위해 정수계획법의 형태를 가지는 제약 충족 문제(constraint satisfaction problem), 동적 프로그래밍 등 많은 형태의 알고리즘이 제시된 바 있다. 그러나 VRP는 외판원 문제(traveling salesman problem)의 확장된 형태이므로, 최적해를 구하는 데에 많은 계산량이 요구되는 NP-hard 문제이다. 따라서 계산 시간을 줄이기 위해 휴리스틱을 사용하는 근사 알고리즘이 연구되어 왔으며[6], 특히 최적 알고리즘과 달리 임무할당 최적화와 경로계획의 최적화를 두 부분으로 나누어 계산시간을 단축시키는 분해 기법이 많은 주목을 받았다[7]. 그러나 위의 연구들은 환경의 불확실성 및 부분적 정보, 그리고 동적으로 환경이 변화할 수 있는 상황을 고려하기 힘들다. 특히 임무 수행 중에 환경이 변화한다면 변화한 환경에 맞게 행동 전략을 빠르게 변경하여야 하지만, 현재의 근사 알고리즘들이 무인기에 실시간으로 적용되기 위해서는 계산시간이 더욱 감소되어야 한다[8].

이러한 단점을 해결하기 위해서 최근에는 무인기의 자율행동에 MDP 및 POMDP를 적용하는 연구가 주목받고 있다. 이러한 의사결정 모델을 이용하였을 경우, 환경 변화의 불확실성 및 행동 주체가 환경에 미치는 영향등을 확률적으로 모델링할 수 있다는 장점이 있다. 또한, 행동 정책을 계산하고 나면, 환경 변화가 일어났을 때 새로운 행동 정책을 다시 계산할 필요가 없다. 이

는 MDP 및 POMDP를 이용하여 얻은 행동 정책은 무인기들이 비행하면서 발생할 수 있는 모든 가능성을 고려한 결과이기 때문이다. 또한 MDP에 비해 POMDP는 부분적 및 부정확한 관찰값을 다룰 수 있다는 장점을 가지기 때문에, 무인기에 장착된 센서의 측정값에 오차가 있을 수 있는 경우를 고려해 더욱 정밀한 행동 정책을 계산할 수 있다.

Bethke 등은 복수 무인기를 위한 정찰 문제에 MDP를 적용하여, 각 무인기들이 연료를 모두 소모하거나 격추되는 일을 줄여 보다 많은 무인기가 정찰임무를 수행할 수 있도록 하는 방법을 제시하였다[2]. 또한 [3]에서는 동적 임무 할당 문제를 POMDP로 모델링하여, 다수의 무인기 간에 임무를 효율적으로 할당할 수 있음을 보였다. 그러나 POMDP의 높은 계산 복잡도로 인해 여러 대의 무인기를 실제 상황에서 기동하기 위해 필요한 임무 할당 및 최적 경로 계산 문제를 함께 해결하는 연구는 아직 초기 단계에 머무르고 있다.

따라서 본 논문에서는 다수의 무인기를 이용하는 정찰 임무 문제에 있어, 임무 할당 및 최적 경로를 함께 고려하도록 POMDP 모델을 구성한다. 또한 POMDP의 계산량을 줄이기 위해, 고수준의 행동 정책을 얻은 다음 이를 따라 무인기가 행동할 수 있도록 제어하는 방법을 제시한다.

3. 순차적 의사결정 모델

3.1 마코프 의사결정 과정(MDP)

MDP[4]는 순차적 의사결정 문제를 위한 가장 일반적인 모델 중 하나로, 의사결정의 주체인 에이전트가 어떠한 행동을 수행하면 행동과 환경의 상태에 따라 그 결과인 보상값(reward)이 주어지게 된다. 따라서 에이전트는 이러한 보상값을 최대로 할 수 있도록 순차적인 행동을 최적화하여야 한다. 수학적으로 MDP는 다음 4개의 항목으로 정의된다:

- 환경 상태의 집합 X
- 수행할 수 있는 행동의 집합 A
- 상태 x 에서 행동 a 를 수행했을 때, 다음 상태가 x' 가 될 전이확률(transition probability) $T(x, a, x') = P(x'|x, a)$
- 상태 x 에서 행동 a 를 수행하고 상태 x' 에 도달했을 때 받는 보상을 정의하는 보상함수(reward function) $R(x, a, x')$
- 감가 인수(discount factor) $\gamma \in [0, 1)$

위 전이확률의 정의에서 알 수 있듯이 MDP 모델에서는 환경의 변화가 마코프 속성(Markov property)을 갖는다고 가정한다. 즉, 다음 상태에 대한 확률은 현재 상태와 현재 수행한 행동에 의해서만 결정되고, 그 이전

의 상태나 행동들의 이력은 영향을 미치지 않는다.

MDP 모델이 정의되면, 에이전트가 앞으로 받을 수 있는 보상값들을 최대화할 수 있는 행동을 알고리즘을 이용하여 계산하게 된다. 이러한 행동 방침을 정책(policy)이라고 한다. 정책은 현재 상태에서부터 행동으로의 함수 $\pi: S \rightarrow A$ 로 정의된다. 즉, 정책 π 를 따르는 에이전트는 상태 s 에서 행동 $\pi(s)$ 를 수행하게 된다. 이러한 에이전트의 행동 규칙은 기본적으로 이전의 모든 상태와 행동에 대한 이력에 기반해야 하지만, 마코프 가정 때문에 현재 상태에만 기반하여도 최적 정책을 표현할 수 있게 된다.

최적 정책을 계산하는 데에 있어 목적 함수로는 주로 보상값의 총합을 이용하게 된다. 그러나 에이전트가 행동할 수 있는 시간이 무한대이면 보상값의 총합도 무한대로 발산할 수 있게 되므로, 보상값의 총합을 유한하게 만들기 위해 주로 감가 보상의 총합(cumulative discounted reward)을 목적 함수로 이용한다. 즉, 미래에 얻게 될 보상값은 현재 얻을 수 있는 즉각적인 보상값보다 중요도가 낮게 된다. 즉, 감가 인수 $\gamma \in [0, 1)$ 를 정의하고, 시간 t 이후의 보상값은 γ^t 을 곱한 만큼 줄어들게 된다. 따라서 에이전트는 감가 보상 총합의 기대값 $\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$ 을 최대화하는 것이 목표가 된다. 여기서 r^t 은 에이전트가 시간 t 에 받은 보상값을 의미한다.

최적 정책 π 는 다음의 Bellman 최적 등식을 만족함이 잘 알려져 있다:

$$V^{\pi}(x) = \sum_{x'} R(x, \pi(x), x') + \gamma T(x, \pi(x), x') V^{\pi}(x').$$

그리고 정책 π 의 Q-value는 다음과 같이 정의된다:

$$Q^{\pi}(x, a) = \sum_{x'} R(x, a, x') + \gamma T(x, a, x') V^{\pi}(x').$$

여기서 가치 함수 $V^{\pi}(x)$ 는 상태 x 로부터 정책 π 를 따라 행동하였을 때 받게 되는 감가 보상 총합의 기대값이다.

MDP는 다항 시간 알고리즘이 존재하여 최적해를 빠르게 계산할 수 있다는 장점이 있지만 현실적인 문제에 바로 적용하기에는 어려움이 있다. MDP는 기본적으로 에이전트가 현재의 상태를 정확하게 알 수 있다는 가정을 전제로 하고 있지만, 실제 문제에 있어서는 현재 상태를 부분적 혹은 부정확하게만 관찰 가능한 경우가 많기 때문이다. 예를 들어 무인기의 경우, 카메라, GPS, 항법센서 등 다양한 센서로부터 얻은 정보에는 어느 정도의 오차가 존재하기 마련이고, 지형 및 방해물에 따라서 그러한 정보를 얻는 것이 불가능한 상황이 생길 수 있다.

3.2 부분관찰 마코프 의사결정 과정(POMDP)

POMDP는 앞서 설명한 MDP의 제약점을 해소하기 위해서 부분적 혹은 부정확한 관찰값을 함께 모델링하

게 된다. 예를 들어 무인기의 현재 위치를 파악하는 데에 있어 센서로부터 얻은 정보는 부정확할 수 있기 때문에, 이러한 정보로부터 실제 위치를 추정하는 동시에 목표 달성을 위해 높은 보상값을 얻을 수 있는 효과적인 행동 정책을 계산해야 한다. POMDP는 MDP의 정의에 다음의 항목들을 추가하여 정의된다.

- 가능한 관찰값들을 정의하는 집합 Z
- 행동 a 를 수행하고 상태가 x' 로 전이되었을 때 관찰값이 z 로 주어질 확률

$$O(x', a, z) = P(z|x', a)$$

- 초기 상태 x 에 대한 확률 $b_0(x) = P(x)$

POMDP에서는 에이전트가 현재 상태를 정확히 알 수 없으므로, 현재 상태에 대한 확률 분포인 확률상태(belief state) b 를 사용하게 된다. 확률상태란 지금까지 수행한 행동과 관측한 관찰값의 이력을 이용하여 추론한 현재 상태에 대한 확률 분포로 정의된다. 즉 확률상태는 $|S|$ 차원의 확률 심플렉스(probability simplex)인 확률상태 공간(belief space) $\Delta(S)$ 에 존재하게 되며, 시간 t 에서의 확률 상태 b_t 는 다음과 같다:

$$b_t(x) = P(x_t = x | z_t, a_{t-1}, z_{t-1}, \dots, a_0, b_0).$$

이러한 확률상태는 바로 이전의 확률상태와 수행한 행동, 그리고 그 결과로 관측한 관찰값을 이용하여 재귀적으로 정의될 수 있다. 즉, $b' = \tau(b, a, z)$ 를 확률상태 b 에서 행동 a 를 수행하고 관찰값 z 를 관측하였을 때 업데이트된 확률상태라 하면 이는 베이즈 정리(Bayes theorem)을 이용하여 다음과 같이 계산할 수 있다:

$$b'(x') = O(x', a, z) \sum_x T(x, a, x') b(x) / P(z|b, a).$$

따라서 POMDP는 확률상태공간 $\Delta(S)$ 을 연속적인 상태 공간으로 가지는 MDP로 생각될 수 있으며, POMDP의 최적 정책은 현재의 확률상태로부터 행동으로의 함수 $\pi | \Delta(S) \rightarrow A$ 로 표현될 수 있다.

하지만 이러한 POMDP의 장점에도 불구하고 POMDP를 실제로 사용하기에는 많은 어려움이 따른다. 그 이유는 POMDP의 최적 정책을 구하는 문제가 MDP에 비해서 매우 높은 계산 복잡도를 가지기 때문이다[9]. 따라서, POMDP를 실제 문제에 사용하기 위해서는 정책을 계산할 때 전체 확률상태 공간을 고려하기 보다는 실제로 도달할 수 있는 확률상태만을 고려하는 점-기반(point-based) 근사 알고리즘[10-13]을 주로 사용한다.

3.3 Factored 표현을 이용한 상태 집합

MDP 및 POMDP를 실제 문제에 적용함에 있어서 가장 큰 어려움 중 하나는 상태를 오직 하나의 변수로 나타내는 flat 표현 방법을 사용하는 경우 가능한 상태의 개수가 너무 많아져 문제를 제대로 표현하기 어렵게

되는 점이다. 이러한 문제점을 해결하기 위해서 상태 및 관찰값을 여러 개의 요소들로 표현하는 factored 표현 방법이 제안된 바 있다[14,15]. 예를 들어, 상태 x 를 $x = (x_1, x_2, \dots, x_n)$ 와 같이 여러 개의 상태 변수를 이용하여 표현하면 전이 확률, 관찰 확률 및 보상 함수 등도 상태 변수들을 이용하여 정의할 수 있어 기존의 방법에 비해서 문제를 더 간결하게 표현할 수 있다. 이러한 방법을 이용한 factored MDP 및 POMDP에서는 문제 표현 및 정책 계산 과정에서의 계산 결과들을 단순한 테이블 대신에 주로 대수 결정 다이어그램(algebraic decision diagrams; ADDs)[16]을 이용하여 표현하게 되고, 이러한 구조를 이용하여 정책 계산 알고리즘을 효율적으로 개선하는 여러 알고리즘들이 제안되었다[14,17,18].

4. POMDP를 이용한 복수 무인기의 정찰 임무 최적화 기법

4.1 문제 정의

본 논문에서 고려하는 복수 무인기의 정찰 임무 문제는 다음과 같다. 무인기의 수는 n 대이며 K 개의 고정된 목표물을 정찰하고 지정된 목표지점에 도달하는 것을 목표로 한다. 또한 임무지역에는 미리 알려진 L 개의 위치에 적 미사일이 존재할 가능성이 있다. 따라서 그림 1에서와 같이 각 무인기는 적 미사일에 격추당하지 않기 위해서 적 미사일의 존재 여부를 확인하고, 이를 회피하며 이동하여 모든 목표물의 정찰을 마치고 목표지점에 도착하여야 한다.

본 문제에서는 무인기의 임무 할당을 관찰하기 위하여 고고도(高高度)를 비행하는 무인기와 저고도(低高度)를 비행하는 무인기로 시스템을 구성한다. 고고도 무인기는 저고도 무인기에 비하여 더 넓은 범위의 미사일 탐지가 가능하며 고고도를 비행하기 때문에 저고도 무인기에 비해 적 미사일에 격추될 가능성이 낮다. 그러나 고고도 무인기는 정찰 범위가 넓은 반면, 그 정확도가 떨어지기 때문에 미사일의 탐지 확률이 저고도 무인기에 비하여 낮고, 또한 무인기들의 임무인 목표물 정찰은 저고도 무인기로만 가능하다고 가정한다. 따라서 저고도 무인기는 적 미사일에 공격당했을 때 격추될 가능성은 높지만 더 정확한 관찰이 가능하며 목표물 정찰

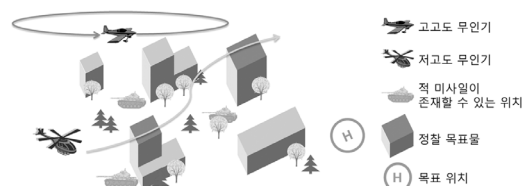


그림 1 복수 무인기의 정찰 임무 문제

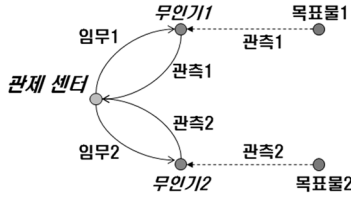


그림 2 관제 센터와 무인기 간의 통신 상황

기능을 수행할 수 있다.

또한 복수의 무인기가 협력을 통하여 성공적으로 임무를 수행하기 위해서는 무인기 간의 동기화가 이루어져야 한다. 예를 들어, 군사적인 목적의 무인기에 위협적인 전장 환경의 경우, 적에게 발각될 위험을 줄이고 목표 지점에서의 임무 달성 확률을 높이기 위해서는 목표 지점이나 중간 도착 지점에 되도록이면 동시에 도달해야 한다[19]. 따라서 본 문제에서는 모든 목표물의 정찰이 끝난 뒤에 목표 지점에 무인기들이 동시에 도착하는 것을 목표로 한다.

각 무인기들은 그림 2와 같이 지상 관제센터를 통하여 서로 관찰 결과를 공유하고 관제센터로부터 임무를 할당 받는다고 가정한다. 이와 같은 가정을 통하여 본 시스템에서는 관제 센터에서 각 무인기들의 관찰 정보를 기반으로 임무를 할당하는 POMDP 모델을 구성하고자 한다.

4.2 POMDP 모델 정의

앞 절에서 소개한 복수 무인기의 정찰 임무 문제를 POMDP로 정의하는 데에 있어서, 실제 환경에서의 무인기 제어에 POMDP를 그대로 적용하기에는 어려움이 따른다. 그 이유는 실제 환경을 정확하게 POMDP 모델로 정의한다면 POMDP 정의에서의 상태, 행동 등이 모두 연속적인 값으로 정의되어야 하기 때문이다. 이 경우 상태 집합 X 및 행동 집합 A 의 크기가 무한대가 되어 POMDP 모델의 정책을 효율적으로 계산하는 것이 불가능해진다. 따라서, 본 논문에서는 복수 무인기의 정찰 임무 문제를 고수준 제어와 저수준 제어로 나누어 해결하고자 한다. 즉, POMDP 모델은 상태, 행동 등을 모두 이산값으로 표현하여 이로부터 무인기들의 고수준 제어를 위한 행동 정책을 얻고, 이 행동 정책을 따라서 행동하도록 무인기를 세밀하게 제어하는 데에는 저수준 제어 방법을 이용한다. 본 절에서는 고수준 제어 정책을 얻기 위한 POMDP 모델을 정의한다.

상태. 상태 x 는 크게 5개의 요소로 나누어진다. 즉,

$$x = \langle s_i, \phi_i, \eta_i, \zeta_k, \xi_l \rangle$$

으로 정의되며, 여기서 s_i 는 i 번째 무인기의 위치를 나타내며, 2차원 격자의 좌표로 표현된다. ϕ_i 는 i 번째 무인기의 생존 여부를 의미하는 Boolean 값이며, η_i 는 i 번째 무인기의 임무 완료 여부를 의미한다. ζ_k 는 k 번

째 정찰지점의 정찰 여부, 그리고 마지막으로 ξ_l 는 i 번째 미사일이 존재하는지의 여부이다.

행동. 무인기의 행동은 동서남북 방향으로 격자 칸을 움직이거나 제자리에 머무를 수 있다고 가정한다. 즉, 행동 집합은 $A = \{\text{East, West, North, South, NoMove}\}$ 로 정의된다. 모든 무인기의 행동은 하나의 시간 단위 내에 완료된다. 그리고 적 미사일과 목표물에 대한 정찰 결과는 매 시간마다 각 행동 이후에 주어진다.

관찰값 및 관찰확률. 관찰값은 무인기의 생존 여부와 적 미사일에 대한 탐지 결과로 이루어진다. 우선 i 번째 무인기의 생존 여부 z_i^ϕ 는 정확하게 관측가능하여 상태 x 의 ϕ_i 값을 정확하게 알 수 있다.

미사일에 대한 탐지 결과에 대한 관찰값은 z_l 로 표현되며, 이는 i 번째 무인기의 탐지 결과를 의미한다. 여기서 적 미사일의 존재 유무는 “존재”, “부존재”, “미확인” 중의 하나로 나타난다. 적 미사일이 관측 가능 범위에 존재하는 경우에, 무인기는 적 미사일이 존재하는지를 관찰할 수 있으나, 센서의 제한된 능력으로 인해서 경우에 따라 잘못된 관찰값을 얻을 수 있다. 따라서 그림 3의 관측 성공 확률에 따라 “존재”와 “부존재”의 관찰값이 주어지게 된다. 또한, 적 미사일이 관측 가능 범위에 존재하지 않는 경우에는 센서로부터 어떠한 정보도 얻을 수 없게 되므로 “미확인”의 관찰값이 주어진다. 또한 고도에 따라 관측 범위 및 성공 확률이 달라진다고 가정하였다. L_l^ξ 를 l 번째 미사일의 위치라고 하면, $\xi_l = 1$ 일 때 저고도 무인기의 경우,

$$P(z_i = \text{존재}) = \begin{cases} 1 & \text{if } \|s_i - L_l^\xi\|_1 = 0 \\ 0.9 & \text{if } \|s_i - L_l^\xi\|_1 = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$P(z_i = \text{부존재}) = \begin{cases} 0.1 & \text{if } \|s_i - L_l^\xi\|_1 = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$P(z_i = \text{미확인}) = \begin{cases} 1 & \text{if } \|s_i - L_l^\xi\|_1 > 1 \\ 0 & \text{otherwise} \end{cases}$$

로 관찰확률이 정의되며, 고고도 무인기의 경우,

$$P(z_i = \text{존재}) = \begin{cases} 0.8 & \text{if } \|s_i - L_l^\xi\|_1 = 0 \\ 0.7 & \text{if } \|s_i - L_l^\xi\|_1 = 1 \\ 0.6 & \text{if } \|s_i - L_l^\xi\|_1 = 2 \\ 0 & \text{otherwise} \end{cases}$$

$$P(z_i = \text{부존재}) = \begin{cases} 0.2 & \text{if } \|s_i - L_l^\xi\|_1 = 0 \\ 0.3 & \text{if } \|s_i - L_l^\xi\|_1 = 1 \\ 0.4 & \text{if } \|s_i - L_l^\xi\|_1 = 2 \\ 0 & \text{otherwise} \end{cases}$$

$$P(z_i = \text{미확인}) = \begin{cases} 1 & \text{if } \|s_i - L_i^\xi\|_1 > 2 \\ 0 & \text{otherwise} \end{cases}$$

와 같이 관찰확률이 정의된다. 관측 범위 내에 여러 개의 적 미사일이 존재하는 경우도 위에서 정의된 관찰확률을 확장하여 표현할 수 있다. 따라서 미사일이 여러 개 존재하면 “존재”의 관찰값을 얻을 확률이 그만큼 높아지게 된다.

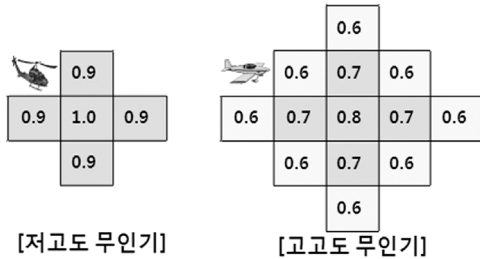


그림 3 무인기의 고도와 미사일 간의 거리에 따른 관찰 확률

전이확률. 전이확률은 다음과 같이 정의된다. 무인기의 다음 위치 s'_i 는 무인기의 현재 위치 s_i 와 행동 a 에 따라 결정론적으로 전이된다. 즉, 무인기의 행동 결과는 항상 정확하다고 가정하였다. 이는 현재 무인기의 활동 영역이 2차원 격자로 표현되었고, 실제 무인기를 이용하였을 때 정확한 제어가 가능했기 때문이다. 단, 무인기가 현재 격추된 상황이라면($\phi_i = 0$), 어떠한 행동을 취하더라도 무인기는 움직이지 않는다. 그리고 무인기가 격자의 가장자리에 위치하고 무인기의 행동이 격자를 벗어나는 방향으로 수행되는 경우에는 무인기는 현재의 위치에 그대로 머무르는 것으로 가정한다.

무인기의 생존 여부 ϕ'_i 는 무인기의 현재 위치 s_i 와 현재의 생존 여부 ϕ_i , 그리고 미사일의 위치 ξ_l 에 따라 확률적으로 전이된다. 무인기가 이미 격추된 상황이라면($\phi_i = 0$), 다시 살아나는 경우가 없다고 가정한다. 따라서 다음과 같이 생존 여부는 0으로 유지된다:

$$P(\phi'_i = 0 | s_i, \phi_i, \xi_l) = 1 \text{ if } \phi_i = 0.$$

현재 i 번째 무인기가 생존해 있는 경우에는, 다음과 같이 무인기와 미사일 간의 거리에 따라서 무인기가 격추될 확률을 정의하였다. 또한 저고도 무인기는 고고도 무인기보다 미사일과의 거리가 가깝기 때문에 격추 확률이 높다고 가정하였다. 그림 4는 저고도 무인기와 고고도 무인기의 격추확률을 보여준다.

또한 수식으로 자세히 표현하면 다음과 같다. $\xi_l = 1$ 일 때 i 번째 무인기가 저고도 무인기인 경우

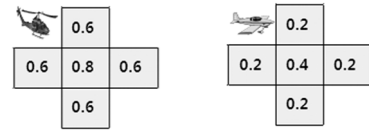


그림 4 무인기의 격추 확률

$$P(\phi'_i = 0) = \begin{cases} 0.8 & \text{if } \phi_i = 1 \text{ and } \|s_i - L_i^\xi\|_1 = 0 \\ 0.6 & \text{if } \phi_i = 1 \text{ and } \|s_i - L_i^\xi\|_1 = 1 \\ 0 & \text{if } \phi_i = 1 \text{ and } \|s_i - L_i^\xi\|_1 > 1 \end{cases}$$

와 같이 격추 확률이 결정되며, 고고도 무인기의 경우는

$$P(\phi'_i = 0) = \begin{cases} 0.4 & \text{if } \phi_i = 1 \text{ and } \|s_i - L_i^\xi\|_1 = 0 \\ 0.2 & \text{if } \phi_i = 1 \text{ and } \|s_i - L_i^\xi\|_1 = 1 \\ 0 & \text{if } \phi_i = 1 \text{ and } \|s_i - L_i^\xi\|_1 > 1 \end{cases}$$

와 같다. 관찰확률과 마찬가지로 격추 범위 내에 여러 개의 미사일이 존재하는 경우, 그만큼 무인기의 격추 확률도 높아지게 된다.

무인기의 임무 완료 여부 η'_i 는 현재의 임무 완료 여부 η_i 와 모든 정찰지점의 정찰 여부 ζ_k , 그리고 현재 무인기의 위치 s_i 및 생존 여부 ϕ_i 에 따라 결정된다. 모든 정찰지점이 정찰되었고, 무인기가 생존해 있으면서 목표 지점에 위치하게 되면 무인기의 임무가 완료된다($\eta'_i = 1$). 무인기가 한번 임무를 완료하고 나면, 임무 완료 여부는 바뀌지 않게 된다. 따라서 η_i 에 대한 전이확률은 다음과 같이 정의된다:

$$P(\eta'_i = 1 | s_i, \phi_i, \zeta_k, \eta_i) = \begin{cases} 1 & \text{if } \eta_i = 1 \\ 1 & \text{if } s_i = g, \phi_i = 1 \text{ and } \zeta_k = 1 \forall k \\ 0 & \text{otherwise.} \end{cases}$$

여기서 g 는 목표 지점의 좌표를 의미한다.

정찰지점의 정찰 여부 ζ'_k 는 저고도 무인기의 현재 위치 s_i 와 생존 여부 ϕ_i , 정찰지점의 현재 정찰 여부 ζ_k 에 따라 결정론적으로 전이된다. k 번째 정찰지점의 위치를 L_k^ζ 라고 하면, L_k^ζ 에 현재 생존해 있는 저고도 무인기가 존재할 경우에 그 지점의 정찰여부가 1로 전이된다. 또한 현재 이미 정찰이 끝난 상황($\zeta_k = 1$)이면 그 정찰지점의 정찰 여부는 변하지 않는다. 즉, 다음과 같이 정의된다:

$$P(\zeta'_k = 1 | s_i, \phi_i, \zeta_k) = \begin{cases} 1 & \text{if } \zeta_k = 1 \\ 1 & \text{if } \exists i. s_i = L_k^\zeta \text{ and } \phi_i = 1 \\ 0 & \text{otherwise.} \end{cases}$$

앞서 설명한 바와 같이 고고도 무인기는 목표물을 정찰할 수 없으므로, 정찰 여부는 고고도 무인기의 상태에 영향을 받지 않는다.

마지막으로 미사일의 존재 여부 ξ_l 은 변하지 않는다.

보상함수. 보상함수는 다음과 같이 정의하였다. 무인기가 격추된 경우 -10의 보상을 주어 격추되는 것을 피하도록 하였다. 또한 임무가 완료되지 않은 경우 매 시간마다 -1의 보상을 주어 최대한 빠른 시간 내에 임무를 완료하도록 한다. 목표물 정찰에 성공한 경우 +50의 보상을 주어 모든 목표물의 정찰을 하도록 하였고, 정찰 완료 후 무인기가 목표 지점에 돌아오는 경우 각각 +10의 보상값을 준다. 특히, 정찰 완료 후 무인기들이 동시에 목표 지점에 도달한 경우 +20의 보상을 주어 동기화된 행동을 유도한다.

결과적으로, 상태를 정의하는 각 상태 변수의 의존 관계를 influence diagram으로 표현하면 그림 5와 같다. 이러한 구조를 가지는 factored MDP 및 POMDP는 3.3절에서 설명한 것처럼 flat MDP 및 POMDP를 계산하는 것보다 더욱 효율적으로 계산할 수 있다.

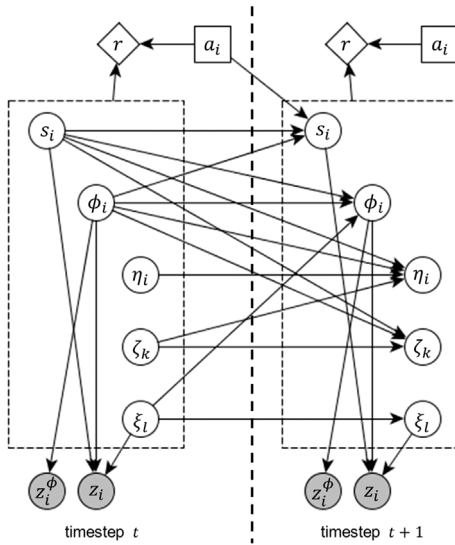


그림 5 복수 무인기 정찰 문제의 influence diagram

4.3 정찰 임무 실험 결과

본 실험의 목적은 POMDP를 이용하여 얻은 고수준 제어 정책의 성능을 평가하고, 관찰확률이 정확하다고 가정한 MDP로부터 계산한 MDP 정책과 성능을 비교하기 위한 것이다. MDP 정책은 미사일 탐지가 정확하다고 가정하기 때문에, 실험 환경인 POMDP 모델로부터 받은 부정확한 탐지 결과를 그대로 믿게 된다. MDP 정책을 POMDP 환경에 적용할 때에는 QMDP[20] 방

법을 사용하였다. 즉, 시뮬레이션 중에는 POMDP 정책과 마찬가지로 관측한 관찰값에 따라서 현재의 확률상태를 업데이트하며, 행동을 결정할 때에는 확률상태와 MDP 정책의 Q-value를 내적하여, 현재 확률상태에서 Q-value를 최대화하는 행동을 수행한다. 각 정책의 성능을 판단하기 위해서는 POMDP 환경에서 각 정책을 이용하여 무인기 정찰을 시뮬레이션하고, 이로부터 얻은 감가 보상의 총합을 이용하였다.

앞서 설명한 대로 정찰 임무 문제에서의 상태 집합은 factored 표현방법으로 구성되었으므로, 이러한 구조를 효율적으로 이용할 수 있는 계산 알고리즘인 SPUDD[14]와 Symbolic HSVI[18]를 각각 factored MDP와 POMDP를 계산하는 데에 사용하였다.

실험에서는 그림 6과 같은 5×5 크기의 격자, 고고도 및 저고도 무인기 각 한대, 그리고 $L=2$, $K=1$ 인 정찰 임무 문제를 이용하였다. 또한 감가 인수로는 0.9를 사용하였다. 표 1은 QMDP 정책과 POMDP 정책을 이용하여 각각 5000번씩 실험한 결과를 보여준다. 여기서 보상값은 50스텝 동안의 감가 보상 총합을 의미한다. 실험 환경 1에서는 앞 장에서 설명한 관찰확률(그림 3)을 사용하였고 실험환경 2에서는 문제를 좀 더 어렵게 만들기 위하여 그림 3에서 0.1을 뺀 값을 사용하였다. 결과에서 알 수 있듯, QMDP 정책보다 POMDP 정책이 평균적으로 더 높은 보상값을 얻었다. 즉, POMDP 정책을 이용했을 때 QMDP에 비해서 무인기가 더 효율적으로 동작하였다. 그러나 보상값은 임의적으로 정해진 보상함수에 의한 값이므로, 무인기의 성능을 좀 더 자세히 평가하기 위해서 임무 성공률 및 정찰 성공률, 그리고 무인기가 격추된 경우의 비율을 함께 계산하였다. POMDP 정책을 이용했을 경우 정찰 목표물을 더 높은 성공률로 정찰할 수 있었으며, 최종적으로 더 높은 확률로 임무를 성공할 수 있음을 알 수 있었다.

QMDP 및 POMDP 정책의 행동을 좀 더 자세히 분석한 결과는 다음과 같다. POMDP 정책의 경우, 그림 7에서처럼 저고도 무인기는 미사일을 피해서 안전한 경로를 이용해 목표위치로 바로 이동하고, 고고도 무인기의 경우 격자의 (1,1) 위치에서 적 미사일을 탐지한다. 그리고 미사일이 없는 경로로 이동하여 목표물을 정찰하고, 목표 위치에 도달한다. 반면, QMDP 정책의 경우, 그림 8에서처럼 고고도와 저고도 무인기 모두 (1,1)의 위치에서 미사일의 존재 유무를 거의 확실하게 판단할 때 까지 오랜 시간동안 머무르게 된다. 미사일 탐지 결과가 정확하다고 판단된 후에 미사일이 없는 경로로 이동하게 된다. 즉, POMDP 정책의 경우 보다 더 효율적인 임무 할당이 이루어짐을 확인할 수 있었다.

실험환경 2의 경우, 관찰확률이 더욱 불확실해졌으

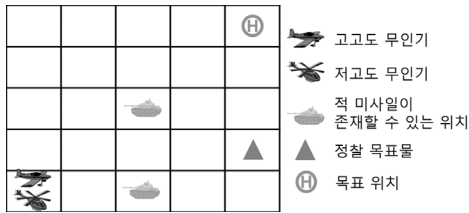


그림 6 정찰 임무 실험 환경

표 1 정찰 임무 실험의 QMDP/POMDP 정책 성능 결과

	실험환경 1		실험환경 2	
	QMDP	POMDP	QMDP	POMDP
보상값	-3.74±0.16	17.14±0.13	-9.95±0	16.90±0
임무성공률	84.52%	97.48%	0%	100%
정찰성공률	86.86%	98.46%	0%	100%
저고도 격추 비율	5.56%	1.34%	0%	0%
고고도 격추 비율	0.24%	1.44%	0%	0%

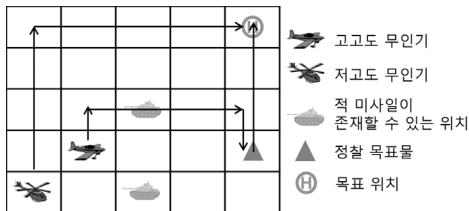


그림 7 POMDP 정책의 행동

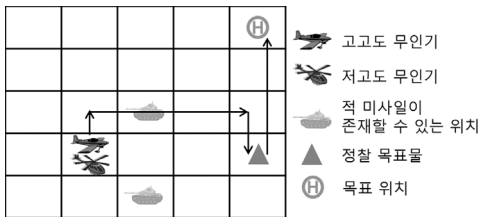


그림 8 QMDP 정책의 행동

로, POMDP 정책의 경우에는 미사일을 탐지하지 않고 미사일이 존재하지 않는 안전한 경로를 택해 이동하는 방법을 택하였다. 따라서 목표 위치까지 도달하는 시간이 오래 걸려 실험환경 1의 경우보다 낮은 보상값을 받았지만 임무 및 정찰의 성공률은 더욱 높았다. QMDP 정책의 경우에는 (1,1)의 위치에서 미사일을 계속 탐지할 뿐, 정찰 및 목표 위치로 이동하지 못하였기 때문에 50스텝 내에 임무와 정찰을 전혀 성공하지 못하는 결과를 보였다.

결과적으로, POMDP 정책을 이용한 경우가 QMDP 정책을 이용한 경우보다 효율적인 임무 할당을 수행하였고, 목표를 빠르게 달성할 수 있는 경로를 찾을 수 있음을 실험을 통해 알 수 있었다.

5. 비행 실험 환경 및 결과

5.1 비행 실험 환경 및 방법

비행 실험은 앞 장에서 POMDP를 이용하여 계산한 고수준 제어 정책이 실제 무인기에 원활하게 적용되는지를 테스트하기 위한 것이다. 비행 실험을 수행하는 데 있어 필요한 비용의 한계 때문에, 무인기의 성능을 평가하는 데 있어서 정량적인 평가보다는 무인기가 계산된 고수준 제어 정책을 잘 따르는지를 분석하는 것으로 실험을 수행하였다.

실험에 사용한 무인기는 쿼드콥터(quadcopter)인 AR Drone[21]으로(그림 9), 최고 18km/h의 속력으로 이동할 수 있고 무게는 420g, 크기는 52.5×51.5cm이다.

비행 실험에 사용된 아키텍처는 그림 10과 같다. 지상 관제센터는 리눅스 기반의 한 대의 컴퓨터로 구성된다. 특히 Robot Operating System(ROS)[22]이라는 오픈 소스 프레임워크를 사용하여 AR Drone의 제어 및 정보 교환을 수행하였다.

AR Drone은 GPS와 같은 센서를 이용한 현재의 위치 정보 제공은 지원하지 않는다. 특히 이는 실내 비행이나 지형 등에 의한 간섭으로 인한 정확한 위치를 알 수 없는 실제적인 임무 환경과 유사한 상황이다. 본 실험에서는 이와 같이 현재 위치 추정이 어려운 상황을 극복하기 위하여 카메라 영상 정보를 이용한 현재 위치 추정 알고리즘을 사용하였다. 무인기가 임무 지역의 지형, 지물 등에 대한 정보를 알고 있다고 가정한다면 장착되어 있는 카메라를 이용하여 영상 정보 처리를 통해 현



그림 9 AR Drone[21]

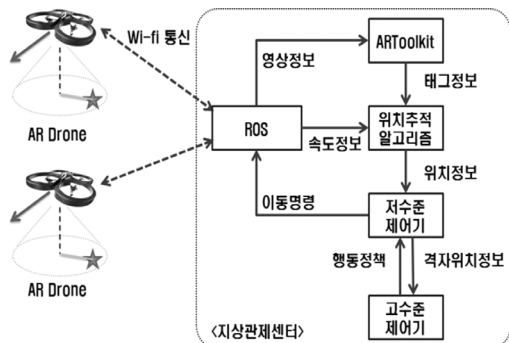


그림 10 비행 실험 구조

재 위치를 추정할 수 있을 것이다. 본 실험에서는 이와 유사한 상황을 가정하기 위해 AR Drone의 임무 지형에 지형, 지물이라 가정한 인식 태그(AR tag)를 설치하였다.

인식 태그는 AR Drone의 하방 카메라를 이용하여 인식되도록 배치하였으며, 이 때 카메라에 특정 태그 영상이 잡힐 경우 인식 라이브러리(AR Toolkit)를 통해 해당 태그에 담긴 무늬를 인식하는 것이 가능하다. 따라서, 각 태그가 배치된 위치정보가 주어진다면 특정 태그를 인식했을 때의 위치를 추정하는 것이 가능하다.

그러나 이러한 영상을 이용한 위치추정은 인식 태그가 하방 카메라의 촬영 가능 범위 안에 존재할 때만 가능하며, 영상 처리에서 발생하는 다양한 오차로 인하여 정확한 위치 추정은 불가능하다. 이러한 단점을 극복하고자 본 실험에서는 태그 인식 정보와 AR Drone의 속도 정보를 기반으로 파티클 필터(particle filter)[23]를 사용하여 현재 위치를 추정하도록 하였다. 이 알고리즘은 이전 시간에 추정된 위치와 새로운 이동/관측 정보를 토대로 새로운 위치를 추정한다. 즉, 현재 시간 t 에서의 무인기의 위치 s_t 에 대한 사후확률을 근사하여 표현하기 위해, 다수의 샘플 s_t^j 을 이용한다. 시간 $t-1$ 에서의 무인기 위치 s_{t-1} 에 대한 샘플 s_{t-1}^j 을 가지고 있다면, 시간 t 에서 수행된 새로운 제어값인 u_t 에 기반하여 새로운 샘플을 사후확률 분포로부터 얻는다. 즉, $s_t^j \sim p(s_t | u_t, s_{t-1}^j)$ 이므로 이를 이용해 s_t^j 을 샘플링한다. 그런 다음 각각의 샘플에 대해서 시간 t 에서의 관측값인 z_t 에 따라 각 파티클의 가중치를 결정한다. 즉, j 번째 파티클의 가중치 w_j 는 $p(z_t | s_t^j)$ 로 정해진다. 마지막으로 각 파티클을 가중치에 따라 재샘플링(resampling)하면 시간 t 에서의 샘플 s_t^j 을 얻을 수 있다.

AR Drone의 제어에 있어서는 파티클 필터를 이용하여 추정된 AR Drone의 위치를 이용하였다. AR Drone의 제어는 저수준 제어와 고수준 제어로 구분할 수 있는데, 고수준 제어는 앞서 설명한 바와 같은 격자에서 학습된 POMDP의 최적 정책을 이용하였다. 저수준 제어는 고수준 제어에서 수행되는 명령을 실제 AR Drone의 이동으로 구현하기 위한 것으로 다음 식과 같은 비례-적분-미분 제어기(proportional-integral-derivative controller)[24]를 사용하였다:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t)$$

위 식에서 e 는 오차, u 는 제어값을 의미한다. 따라서 제어값은 각각 오차값, 오차값의 적분, 오차값의 미분에 비례한다. 또한 K_p 는 비례항(proportional gain)을, K_i 는 적분항(integral gain)을, K_d 는 미분항(derivative

gain)을 나타낸다. 비례항은 현재 상태에서 오차의 크기에 비례하게 제어작용을 하도록 하고, 적분항은 정상상태(steady-state) 오차를 없애는 작용, 미분항은 제어작용에 있어 안정성을 향상시키기 위한 것이다.

5.2 비행 실험 결과

AR Drone의 비행 실험 환경은 그림 11과 같이 구성되었다. 4.3절에서의 실험 환경과 마찬가지로 5×5 크기의 격자, 고고도 및 저고도 무인기 각 한대, 그리고 $L=2$, $K=1$ 인 정찰 임무 문제를 이용하였고, 격자 하나의 크기는 $1.2\text{m} \times 1.2\text{m}$ 이었다.

본 실험에서는 파티클 필터에서의 전이확률 $p(s_t | u_t, s_{t-1})$ 를 $s_{t-1} + u_t$ 를 평균으로 가지고 $0.5^2 I$ 를 공분산 행렬로 가지는 정규분포로, 관측확률 $p(z_t | s_t^j)$ 을 s_t^j 를 평균, $0.05^2 I$ 를 공분산 행렬로 가지는 정규분포로 가정하였다. 즉, 무인기의 이동에는 50cm를 표준편차로 가지는 에러가 존재하고, 무인기의 실제 위치와 AR tag를 이용하여 추정된 위치 사이에는 5cm를 표준편차로 가지는 에러가 존재한다고 가정하였다.

적 미사일이 존재할 수 있는 위치와 정찰 목표물이 그림 11과 같이 구성된 경우, 고고도 무인기는 먼저 (2,3)의 위치에 가서 두 개의 적 미사일 존재 여부를 확인하는 행동을 보였다. 그 결과 (3,2)에 적 미사일이 없다고 판단될 때 무인기들의 비행경로는 그림 12와 같았다. 즉, 저고도 무인기는 (3,2)에 미사일이 존재할 확률이 낮다고 판단하여 정찰 목표물의 위치로 곧바로 이동하였다. 반대로, 적 미사일이 (3,2)에 존재하고 (3,4)에 없다고 판단될 경우의 무인기의 비행 경로는 각각 그림 13과 그림 14와 같았다. 고고도 무인기의 경우 미사일 존재 여부를 확인한 다음 목표위치로 이동하였고, 저고도 무인기의 경우는 미사일이 존재한다고 판단한 (3,2) 위치를 피하여 정찰 목표물의 위치로 이동하였다.

결과적으로, POMDP를 이용하여 무인기의 고수준 행동 정책을 계산한 결과, 환경의 변화에 따라 동적으로 적절한 임무를 각 무인기에게 할당하고 효과적인 비행

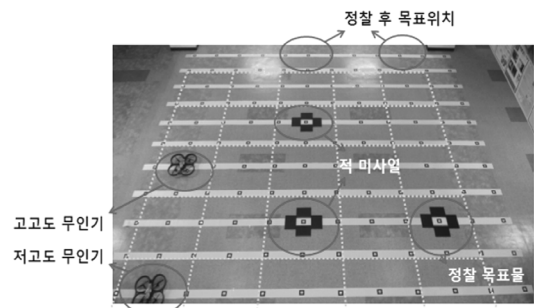


그림 11 비행 실험 환경

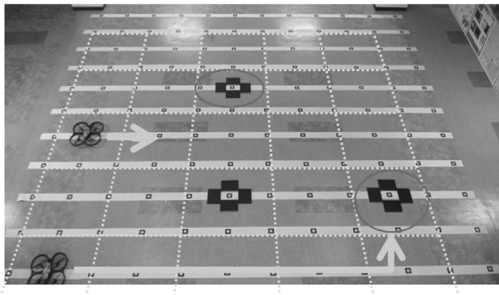


그림 12 (3,2)에 적 미사일이 없을 때의 비행 경로

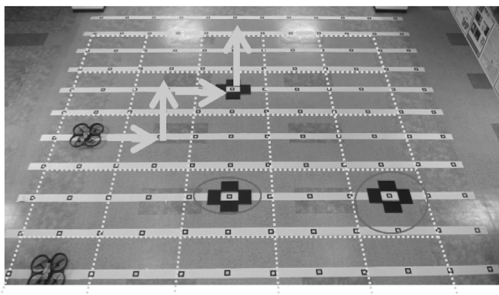


그림 13 (3,4)에 적 미사일이 없을 때 고고도 무인기의 경로

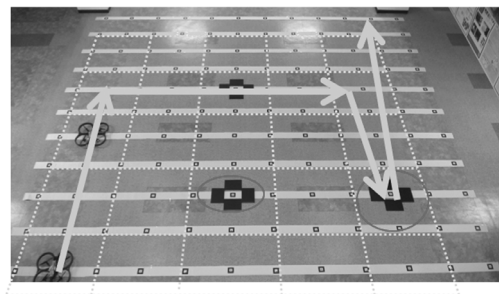


그림 14 (3,4)에 적 미사일이 없을 때 저고도 무인기의 경로

경로를 계산함으로써 임무를 효율적으로 수행할 수 있음을 실제 비행 실험을 통해 확인하였다.

6. 결론

본 논문에서는 복수의 무인기를 이용한 정찰 임무 수행 문제를 POMDP를 이용하여 모델링하였다. 또한 이로부터 얻은 행동 정책을 이용하면 환경의 변화에 따라서 동적으로 적절한 임무를 각 무인기에게 할당하고 효과적인 비행 경로를 계산함으로써 정찰 임무를 효율적으로 수행할 수 있음을 보였다. 특히 환경으로부터 얻은 관찰값이 부정확한 경우를 고려함으로써, 이를 고려하지 않은 MDP를 이용했을 때 보다 높은 성능을 얻을 수 있음을 실험을 통하여 확인하였다.

또한 POMDP로부터 얻은 고수준의 행동 정책을 AR Drone을 이용한 실제 비행 실험에 적용하였으며, POMDP 정책을 통한 효과적인 제어가 가능하였다. 특히 고수준 제어기와 저수준 제어기의 계층적 구성을 통하여 POMDP 문제를 간단하게 표현함으로써 효율적인 최적화를 수행할 수 있었다.

현재 고려하고 있는 후속 연구는 다음과 같다. 첫째로, 본 논문에서는 관제센터와 무인기 간의 교신이 원활하다는 가정하에, 여러 대의 무인기의 의사 결정을 하나의 관제 센터에서 처리하는 모델을 이용하였다. 그러나 실제로 복수 무인기를 사용하는 데에 있어서, 적의 레이더 탐지를 피해야 하거나 통신 장비 능력의 제한 등으로 원활한 교신이 어려운 경우가 많다. 이러한 상황에서는 각 무인기가 개별적으로 의사 결정을 수행할 수 있는 분산 제어 기법이 필요하다. 따라서 교신이 불가능하거나 제약이 있는 상황에서의 자율 행동을 위한 분산(decentralized) POMDP[25-27]를 복수 무인기에 적용하는 연구를 진행할 계획이다. 둘째, 본 논문에서는 POMDP 정책을 효율적으로 계산하기 위하여 실제 환경을 보다 간단하게 만든 환경에 대한 고수준 정책을 구하고 이를 저수준 제어에 사용하는 방법을 이용하였지만, 실제 환경에서 보다 효율적이고 세밀한 제어를 수행하기 위해서는 실제 환경을 보다 더 정확하게 모델링하고, 이로부터 POMDP 정책을 효율적으로 계산할 수 있어야 한다. 이를 위해서는 상태, 관찰값 및 행동이 연속적인 값으로 표현되는 continuous POMDPs[28]를 사용할 수 있을 것이다. 마지막으로, 문제의 크기로 인하여 POMDP 정책의 계산이 어려운 경우, 매 의사결정 순간마다 현재 처한 상황에 대해 필요한 지역적인 정책만을 계산하여 사용하는 온라인 계획수립 알고리즘[29]을 복수 무인기에 적용하는 연구를 진행할 계획이다.

참 고 문 헌

- [1] B. Grocholsky et al., "Cooperative air and ground surveillance - A scalable approach to the detection and localization of targets by a network of UAVs and UGVs," *IEEE Robotics & Automation Magazine*, vol.13, no.3, pp.16-26, 2006.
- [2] B. Bethke, J. How, and J. Vian, "Group health management of UAV teams with applications to persistent surveillance," in *IEEE American Controls Conference*. 2008.
- [3] S.A. Miller, Z.A. Harris, and E.K.P. Chong, "A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking," *Eurasip Journal on Advances in Signal Processing*, 2009.
- [4] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1994, New York: John Wiley and Sons, Inc.

- [5] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra, *Planning and acting in partially observable stochastic domains*, Artificial Intelligence, vol.101, no.1-2, pp.99-134, 1998.
- [6] G. Laporte, "What you should know about the vehicle routing problem," *Naval Research Logistics*, vol.54, no.8, pp.811-819, 2007.
- [7] A. Richards et al., "Coordination and control of multiple UAVs," in *Proc. of AIAA Guidance, Navigation, and Control Conference*. 2002.
- [8] M. Alighanbari, Y. Kuwata, and J.P. How, "Coordination and Control of Multiple UAVs with Timing Constraints and Loitering," in *Proc. of American Control Conference*. 2003.
- [9] C.H. Papadimitriou and J.N. Tsitsiklis, "The complexity of Markov decision-processes," *Mathematics of Operations Research*, vol.12, no.3, pp.441-450, 1987.
- [10] J. Pineau, G. Gordon, and S. Thrun, "Anytime point-based approximations for large POMDPs," *Journal of Artificial Intelligence Research*, vol.27, pp.335-380, 2006.
- [11] T. Smith and R. Simmons, "Heuristic search value iteration for POMDPs," in *Proc. of UAI*. 2004.
- [12] T. Smith and R. Simmons, "Point-based POMDP algorithms: Improved analysis and implementation," in *Proc. of UAI*. 2005.
- [13] M.T.J. Spaan and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs," *Journal of Artificial Intelligence Research*, vol.24, pp.195-220, 2005.
- [14] J. Hoey et al., "SPUDD: Stochastic planning using decision diagrams," in *Proc. of UAI*. 1999.
- [15] C. Boutilier and D. Poole, "Computing optimal policies for partially observable decision processes using compact representations," in *Proc. of AAAI*. 1996.
- [16] R.I. Bahar et al., "Algebraic decision diagrams and their applications," in *Proc. of the IEEE/ACM Int. Conf. on Computer-Aided Design*. 1993.
- [17] E. Hansen and Z. Feng, "Dynamic programming for POMDPs using a factored state representation," in *Proc. of ICAPS*. 2000.
- [18] H.S. Sim et al., "Symbolic heuristic search value iteration for factored POMDPs," in *Proc. of AAAI*. 2008.
- [19] T.W. McLain and R.W. Beard, "Coordination variables, coordination functions, and cooperative-timing missions," *Journal of Guidance Control and Dynamics*, vol.28, no.1, pp.150-161, 2005.
- [20] M.L. Littman, A.R. Cassandra, and L.P. Kaelbling, "Learning policies for partially observable environments: scaling up," in *ICML*. 1995.
- [21] <http://ardrone.parrot.com/parrot-ar-drone/usa/>.
- [22] <http://www.ros.org/wiki/>.
- [23] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 2005, MIT Press.
- [24] Y. Li, K.H. Ang, and C.C.Y. Chong, "PID control system analysis and design - Problems, remedies, and future directions," *IEEE Control Systems Magazine*, vol.26, no.1, pp.32-41, 2006.
- [25] M.T.J. Spaan, G.J. Gordon, and N. Vlassis, "Decentralized planning under uncertainty for teams of communicating agents," in *Proc. of AAMAS*. 2006.
- [26] E.A. Hansen, D.S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *Proc. of AAAI*. 2004.
- [27] D.S. Bernstein, E.A. Hansen, and S. Zilberstein, "Bounded policy iteration for decentralized POMDPs," in *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*. 2005.
- [28] J.M. Porta et al., "Point-based value iteration for continuous POMDPs," *Journal of Machine Learning Research*, vol.7, pp.2329-2367, 2006.
- [29] S. Ross et al., "Online planning algorithms for POMDPs," *Journal of Artificial Intelligence Research*, vol.32, pp.663-704, 2008.



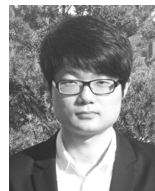
김 동 호

2003년 KAIST 전산학과, 수학과 학사
2011년 KAIST 전산학과 박사. 2011년~2012년 KAIST 전산학과 박사후연구원. 2012년~현재 University of Cambridge 박사후연구원. 관심분야는 인공지능, 기계학습, 강화학습



이 재 송

2005년 KAIST 전산학과 학사. 2012년 KAIST 전산학과 석사. 2012년~현재 KAIST 전산학과 박사과정. 관심분야는 인공지능, 기계학습, 강화학습



최 재 득

2008년 Postech 컴퓨터공학과 학사. 2009년 KAIST 전산학과 석사. 2009년~현재 KAIST 전산학과 박사과정. 관심분야는 인공지능, 기계학습, 강화학습



김 기 응

1995년 KAIST 전산학과 학사. 1998년 브라운대학교 전산학과 석사. 2001년 브라운대학교 전산학과 박사. 2001년~2003년 삼성 SDS 책임연구원. 2004년~2005년 삼성종합기술원 책임연구원. 2005년~현재 KAIST 전산학과 조교수. 관심분야는 인공지능, 기계학습, 강화학습