論文 DOI:http://dx.doi.org/10.5139/JKSAS.2012.40.3.215

에피소드 매개변수 최적화를 이용한 확률게임에서의 추적정책 성능 향상

곽동준*, 김현진**

Improvements of pursuit performance using episodic parameter optimization in probabilistic games

Dong Jun Kwak* and H. Jin Kim**

ABSTRACT

In this paper, we introduce an optimization method to improve pursuit performance of a pursuer in a pursuit-evasion game (PEG). Pursuers build a probability map and employ a hybrid pursuit policy which combines the merits of local-max and global-max pursuit policies to search and capture evaders as soon as possible in a 2-dimensional space. We propose an episodic parameter optimization (EPO) algorithm to learn good values for the weighting parameters of a hybrid pursuit policy. The EPO algorithm is performed while many episodes of the PEG are run repeatedly and the reward of each episode is accumulated using reinforcement learning, and the candidate weighting parameter is selected in a way that maximizes the total averaged reward by using the golden section search method. We found the best pursuit policy in various situations which are the different number of evaders and the different size of spaces and analyzed results.

초 록

본 논문에서는 추적-회피 게임에서 추적자의 추적성능을 향상시키기 위한 최적화 기법을 소개한다. 제한된 공간상에서 추적자는 도망자를 빠른 시간 내에 찾아내고 잡기 위해 확률맵을 생성하고 그 확률정보를 토대로 탐색한다. 추적자는 기존 global-max와 local-max의 장점을 취한 hybrid 추적방식을 사용하는데 이 추적방식은 global-max와 local-max 성향을 조절하는 가중치를 갖는다. 따라서 상황별 최적의 가중치를 찾기 위해 에피소드 매개변수 최적화 알고리즘을 제안하였다. 이 알고리즘은 가중치에 대한 다수의 추적-회피 게임 에피소드를 반복적으로 수행하는 동안 강화학습을 통해 보상을 누적한 후 해당 가중치의 평균보상을 최대화 하는 방향으로 황금분할법을 사용하여 최적의 가중치를 찾는다. 이 최적화 기법을 이용하여 여러 상황별 최적 추적정책을 찾기 위해 도망자 수와 공간의 크기를 변화시켜가며 각각 최적화를 수행하였고 그 결과를 분석하였다.

Key Words : Multi-agent system(멀티 에이전트 시스템), Pursuit-Evasion Game(추적-회피 게임), Parameter Optimization(매개변수 최적화), Reinforcement Learning (강화학습)

† 2011년 10월 17일 접수 ~ 2012년 1월 26일 심사완료
* 정회원, 서울대학교 기계항공공학부
** 정회원, 서울대학교 기계항공공학부
교신저자, hjinkim@snu.ac.kr,
서울시 관악구 관악로 599번지

Ⅰ. 서 론

감시정찰과 같은 단일로봇만으로 수행하기 힘든 임무에 대해 다중로봇은 임무시간을 단축할 수 있고 임무수행 성능을 높일 수 있기 때문에

이와 관련된 연구가 많이 수행되고 있다. 다중로봇의 감시정찰 임무와 관련하여 본 논문에서는 추적자는 최단 시간에 도망자를 잡기 위해 노력하고 반대로 도망자는 최대한 추적자들로부터 회피하려 하는 추적-회피 게임에 대해서 다루고자 한다. 추적-회피 게임의 기본 이론은 R. Isaacs[1]에 의해 처음 소개되었고, R. Vidal 등[2]은 추적-회피 게임을 확률적인 관점으로 접근하였다. L. Schenato등[3]은 센서 네트워크를 추적-회피 게임과 연동하여 연구를 수행하였고, 곽동준 등[4,5]은 확률기반 추적-회피 게임에서 다수의 추적자와 하나의 도망자에 대해서 추적자들의 수와 공간 크기의 변화에 대한 연구 및 추적자의 추적방식과 도망자의 회피방식 간 경향성을 분석하였다. 앞선 확률기반 추적-회피 게임 연구에서 추적자들은 local-max와 global-max라 불리는 추적방식을 사용하는데 local-max 추적방식은 확률맵에서 추적자 자신 근방에서 가장 확률이 높은 곳을 찾고, global-max 추적방식은 확률맵 전체로부터 확률이 가장 높은 지점을 탐색한다. 이 두 가지 방식은 각각 장단점이 존재하는데 local-max 추적방식은 global-max 추적방식에 비해 하나의 도망자를 잡는데 시간이 오래 걸리지만 서로 멀리 떨어져 탐색하여 다수의 도망자에는 유리할 것으로 예상되었고, global-max 추적방식의 경우 빠른 시간 내에 한 기의 도망자를 잡아냈지만 모든 추적자들이 한 곳으로 집중되어 다수의 도망자에는 적합하지 않은 추적방식으로 생각되었다. 따라서 본 논문에서 이 두 추적방식의 장점을 취한 hybrid 추적방식[6]을 고려하였고 추적-회피 게임 시나리오별 최적의 추적정책을 찾고자 한다.

(a)

(b)

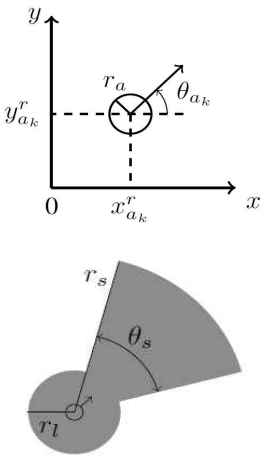

Fig. 1. (a)Agent model, (b)Sensor model ($r_a = 3.5\,\mathrm{cm}$, $r_l = 15.6\,\mathrm{cm}$, $r_s = 60\,\mathrm{cm}$, and $\theta_s = 60^\circ$)

II. 추적-회피 게임 설정

본 절에서는 추적-회피 게임의 기본 설정에 대해 소개하고자 한다.

2.1 가정

- N_P 추적자들과 N_E 도망자들은 $N_{ROW} \times N_{COL}$ 의 분할된 셀 $\boldsymbol{X}$ 상에서 움직인다.
- 추적자들은 매 순간 센서를 이용하여 도망자를 탐지한다.
- 센서 모델은 거짓양성의 확률을 가지고 있다.
- 모든 도망자들이 잡히면 게임은 종료된다.

2.2 에이전트의 동적모델

Figure 1(a)와 같이 에이전트 a_k는 $N_{GEO} \times N_{GEO}$ 공간 내에서 위치 $\boldsymbol{x}^r_{a_k} = (x^r_{a_k}, y^r_{a_k}) \in \boldsymbol{X}^r$와 방향각 θ_{a_k}를 가진다. 이산공간 상에서 에이전트 a_k의 위치 $\boldsymbol{x}^r_{a_k}$는 $\boldsymbol{x}_{a_k} \in \boldsymbol{X}$ 로 다음과 같이 매핑 할 수 있다.

$$x_{a_k} = \left\lceil \frac{x^r_{a_k} N_{ROW}}{N_{GEO}} \right\rceil, y_{a_k} = \left\lceil \frac{y^r_{a_k} N_{COL}}{N_{GEO}} \right\rceil \quad (1)$$

여기서 $\lceil . \rceil$ 은 천장함수(ceiling function)를 의미한다. 추적자 p_i는 위치 $\boldsymbol{x}_{p_i} \in \boldsymbol{X}$와 방향각 θ_{p_i}, 도망자 e_j는 위치 $\boldsymbol{x}_{e_j} \in \boldsymbol{X}$와 방향각 θ_{e_j}에 대한 정보를 가지고 있다. 각 에이전트들은 Fig. 1(b)와 같은 센서를 가지고 확률맵을 생성하거나 다른 에이전트의 움직임을 탐지하고, 식 (2)와 같이 움직인다.

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= w \end{aligned} \quad (2)$$

여기서 $v = 13$, $w \in [-5.43, 5.43]$이고, 에이전트를 나타내는 첨자는 생략하였다.

2.3 제어법칙

에이전트 a_k의 방향각 θ_{a_k}와 가고자 하는 방향 θ_d와의 각도 차 θ_e에 비례게인 K_p를 곱함으로써 제어 입력 w는 다음과 같이 정할 수 있다.

$$w = K_p \theta_e = K_p(\theta_d - \theta_{a_k}) \quad (3)$$

2.4 충돌회피

에이전트 간 충돌회피[7]는 가장 높은 우선순위

를 가진다. 에이전트 a_i는 다른 에이전트 a_j와 최소 반경 r_r을 유지하기 위해 다음과 같이 가고자 하는 방향 $\boldsymbol{d}_{a_i}$을 정한다.

$$\boldsymbol{d}_{a_i}=-\sum_{\boldsymbol{x}_{a_j}\in C(\boldsymbol{x}_{a_j})}\frac{\boldsymbol{x}_{a_j}-\boldsymbol{x}_{a_i}}{|\boldsymbol{x}_{a_j}-\boldsymbol{x}_{a_i}|} \tag{4}$$

여기서 $C(\boldsymbol{x})=\{\boldsymbol{x}\in\boldsymbol{X}:|\boldsymbol{x}-\boldsymbol{x}_{a_i}|<r_{r_Z}\}$ 이고, $r_{r_Z}=\lceil r_r\sqrt{(N_{ROW}/N_{GEO})^2+(N_{COL}/N_{GEO})^2}\rceil$ 이다. 추적-회피 게임 상에서 추적자가 도망자를 잡게 되면, 도망자는 장애물이 되고 각각의 에이전트는 장애물에 대해서도 충돌회피가 이루어진다.

III. 확률기반 추적-회피 게임

본 절에서는 추적자가 제한된 공간상에서 최대한 빨리 도망자를 찾고 잡기위한 확률맵 생성 및 업데이트 방식과 추적자와 도망자의 추적 및 회피방식을 소개한다.

3.1 확률맵

추적자들은 각자의 센서정보를 공유하여 $N_{ROW}\times N_{COL}$의 셀 $\boldsymbol{x}_c\in\boldsymbol{X}$로 이루어진 확률맵을 생성한다. 그리고 확률맵을 매시간 t마다 업데이트 한다. Y_t를 시간 t까지의 센서데이터 집합으로 정의하면 Y_t는 관측된 셀들의 좌표와 도망자의 탐지에 대한 2진의 정보로 구성된다. 도망자가 시간 τ에서 셀 $\boldsymbol{x}_c$상에 있을 조건부 확률은 $p_e(\boldsymbol{x}_c,\tau|Y_t)$로 나타낸다. 다음은 확률맵이 생성되는 과정을 나타낸다.

Step 1. 초기에 각각의 셀 $\boldsymbol{x}_c\in\boldsymbol{X}$는 균등 확률 분포를 가진다. 후에 확률분포는 센서정보에 따라 업데이트 된다.

Step 2. 이전 시간 $t-\Delta t$의 확률 맵 $p_e(\boldsymbol{x}_c,t-\Delta t|Y_{t-\Delta t})$은 현재 업데이트된 새로운 센서정보 Y_t에 의해 다음과 같이 변경된다.

$$p_e(\boldsymbol{x}_c,t-\Delta t|Y_t)=\begin{cases}0.9 & \text{if } \boldsymbol{x}_c \text{ is observed and there is an evader in } \boldsymbol{x}_c\\ 0 & \text{if } \boldsymbol{x}_c \text{ is observed and there is no evader in } \boldsymbol{x}_c\\ p_e(\boldsymbol{x}_c,t-\Delta t|Y_{t-\Delta t}) & \text{if } \boldsymbol{x}_c \text{ is not observed}\end{cases} \tag{5}$$

여기서 거짓양성(false positive)의 가능성을 반영하기 위해 추적자들은 오직 90%의 도망자의 존재에 대한 확률만 신뢰하도록 하였다.

Step 3. 시간 t에서 센서정보 Y_t에 의해 도망자가 셀 $\boldsymbol{x}_c$에 존재할 확률은 식 (6)과 같다.

$$p_e(\boldsymbol{x}_c,t|Y_t)=\frac{p_e(\boldsymbol{x}_c,t-\Delta t|Y_t)}{\sum_{\boldsymbol{x}_c\in\boldsymbol{X}}p_e(\boldsymbol{x}_c,t-\Delta t|Y_t)} \tag{6}$$

Step 4. Step 2로 가서 반복

3.2 확률맵 업데이트

추적자들은 다음 시간 $t+\Delta t$에서 셀 $\boldsymbol{x}_c$상에 도망자가 있을 확률을 업데이트 한다. $s(\boldsymbol{x},\boldsymbol{x}')$를 $\boldsymbol{x}$와 $\boldsymbol{x}'$사이를 이동하는데 걸리는 시간이라고 하면, $N(\boldsymbol{x})=\{\boldsymbol{x}'\in\boldsymbol{X}:s(\boldsymbol{x},\boldsymbol{x}')=1\}$는 1초 안에 이동할 수 있는 주위의 셀들로 정의한다. p_{move}를 도망자가 이동할 확률이라 하면 주어진 측정정보 Y_t에 대한 시간 $t+\Delta t$에서 셀 $\boldsymbol{x}_c$에 도망자가 존재할 확률은 다음과 같이 정의할 수 있다.

$$\begin{aligned}p_e(\boldsymbol{x}_c,t+\Delta t|Y_t)=&(1-p_{move})p_e(\boldsymbol{x}_c,t|Y_t)\\&+\sum_{\boldsymbol{x}_c'\in N(\boldsymbol{x}_c)}p_{move}p_e(\boldsymbol{x}_c',t|Y_t)\end{aligned} \tag{7}$$

3.3 추적방식

3.3.1 Local-max 추적방식

추적자 p_i는 확률맵 상에서 자신의 주위 반경 $r_{l_Z}=\lceil r_l\sqrt{(N_{ROW}/N_{GEO})^2+(N_{COL}/N_{GEO})^2}\rceil$ 내에 셀의 집합 $L(\boldsymbol{x})=\{\boldsymbol{x}_c\in\boldsymbol{X}:|\boldsymbol{x}-\boldsymbol{x}_c|<r_{l_Z}\}$ 중에서 가장 높은 확률을 가지는 위치를 다음과 같이 찾는다.

$$\boldsymbol{x}_{\max}=\arg\max_{\boldsymbol{x}_c'\in L(\boldsymbol{x}_{p_i})}p_e(\boldsymbol{x}_c',t+\Delta t|Y_t) \tag{8}$$

3.3.2 Global-max 추적방식

추적자 p_i는 전체 확률맵에서 확률이 최대가 되는 지점을 찾는다. 그러나 만약 추적자의 위치와 최대 확률 지점 사이가 너무 멀리 떨어져 있으면 추적자가 최대 확률 지점으로 이동하는 동안 최대 확률 지점의 확률 값이 급격하게 감소할 수 있다. 따라서 확률맵에 현재 추적자 위치와의 거리에 대한 가중치를 주어 다음과 같이 시간 t에서의 최대 확률 지점을 정의한다.

$$\boldsymbol{x}_{\max}=\arg\max_{\boldsymbol{x}_c\in X}\frac{p_e(\boldsymbol{x}_c,t+\Delta t|Y_t)}{|\boldsymbol{x}_c-\boldsymbol{x}_{p_i}|} \quad (9)$$

3.3.3 Hybrid 추적방식

곽동준 등[4]의 연구에서 앞서 언급한 두 가지 방식의 추적방식에 대해 공간의 크기와 추적자의 수의 변화를 비교한 결과, 하나의 도망자에 대해 global-max 추적방식이 local-max 추적방식에 비해 빠른 시간 안에 도망자를 잡아 좋은 성능을 보여주었지만, 하나의 추적자가 도망자를 발견했을 경우에 도망자가 위치에 대한 확률 값이 너무 크기 때문에 모든 추적자들이 그 지점으로 집중되는 약점을 보여주었다. local-max 추적방식을 사용할 경우에는 추적자들이 집중되지 않고 서로 떨어져서 탐색을 하는 경향으로부터 다수의 도망자에 대해서 좋은 성능을 보여줄 것으로 예상할 수 있었지만 global-max 추적방식에 비해 빠른 시간 안에 도망자를 잡아내지 못했다. 따라서 이 두 추적방식의 장점을 취한 hybrid 추적방식을 다음과 같이 정의한다.

$$p_{e,h}(\boldsymbol{x}_c,t+\Delta t|Y_t)=\begin{cases}w_l w_h(t)p_e(\boldsymbol{x}_c,t+\Delta t|Y_t)\\ \quad+(1-w_l)\dfrac{p_e(\boldsymbol{x}_c,t+\Delta t|Y_t)}{|\boldsymbol{x}_c-\boldsymbol{x}_{p_i}|} & \text{if } |\boldsymbol{x}_c-\boldsymbol{x}_{p_i}|<r_{l_Z}\\ (1-w_l)\dfrac{p_e(\boldsymbol{x}_c,t+\Delta t|Y_t)}{|\boldsymbol{x}_c-\boldsymbol{x}_{p_i}|} & \text{if } |\boldsymbol{x}_c-\boldsymbol{x}_{p_i}|\geq r_{l_Z}\end{cases}$$

$$\boldsymbol{x}_{\max}=\arg\max_{\boldsymbol{x}_c\in X}p_{e,h}(\boldsymbol{x}_c,t+\Delta t|Y_t) \quad (10)$$

여기서 가중치 w_l은 local-max 추적방식과 global-max 추적방식의 영향력을 조절하기 위한 가중치로써 다음 절의 강화학습으로부터 정해진다. 만약 도망자가 추적자의 센서범위 내에 존재하면 그 위치의 확률 값이 너무 커지는 것으로 인한 local-max 영향력의 감소로 인해 다음의 가중치 함수 $w_h(t)$를 도입하였다.

$$w_h(t)=\begin{cases}\dfrac{\max_{\boldsymbol{x}_c\in X}p_e(\boldsymbol{x}_c,t+\Delta t|Y_t)}{\overline{p_e}}(=W_h) & \text{if } W_h>1\\ 1 & \text{if } W_h\leq 1\end{cases} \quad (11)$$

여기서 $\overline{p_e}$는 전체 확률맵의 평균이다. 따라서 목표지점 $\boldsymbol{x}_{\max}=(x_{\max},y_{\max})$에 대한 방향각은 다음과 같다.

$$\theta_d=\text{atan2}\left(\frac{y_{\max}-y_{p_i}}{x_{\max}-x_{p_i}}\right) \quad (12)$$

3.4 지능적인 회피방식

도망자는 추적자들로부터 지능적으로 회피하기 위해 추적자의 위치와 확률맵 정보를 알고 있다고 가정하였다. 도망자는 추적자의 위치정보를 토대로 퍼텐셜 함수 기법[8]을 이용하여 각 추적자들에 반발 퍼텐셜을 생성하여 도망자가 추적자들로부터 멀리 떨어져 있게 하였고, 확률맵 정보를 토대로 추적자들의 예측에 미리 대응하도록 하였다. 따라서 도망자의 의사결정 함수 $PF_e(\boldsymbol{x}_c,t)$는 다음과 같이 정하였다.

$$PF_e(\boldsymbol{x}_c,t)=\sum_{p_i=1}^{N_p}\frac{A}{2}\left(\frac{1}{|\boldsymbol{x}_c-\boldsymbol{x}_{p_i}|}-\frac{1}{\rho_0}\right)^2+Bp_e(\boldsymbol{x}_c,t+\Delta t|Y_t) \quad (13)$$

여기서 더 나은 회피를 위해 강화학습을 이용하여 가중치 A와 B를 결정할 수 있다.
도망자는 의사결정 함수 $PF_e(\boldsymbol{x}_c,t)$와 local-min 방식으로부터 목표지점 $\boldsymbol{x}_{\min}$과 방향각 θ_d를 다음과 같이 정할 수 있다.

$$\boldsymbol{x}_{\min}=\arg\min_{\boldsymbol{x}_c'\in L(\boldsymbol{x}_{e_j})}PF_e(\boldsymbol{x}_c',t) \quad (14)$$
$$\theta_d=\text{atan2}\left(\frac{y_{\min}-y_{e_j}}{x_{\min}-x_{e_j}}\right)$$

IV. 에피소드 매개변수 최적화

본 절에서는 3.3절에서 정의한 최적의 추적정책을 찾기 위해 강화학습 기반의 에피소드 매개변수 최적화 알고리즘을 소개한다.

4.1 강화학습

최적의 hybrid 추적방식을 찾기 위해 강화학습 문제[9]로 정의하면 임의의 가중치 w_l에 대한 hybrid 추적방식의 기대값을 통해 성능을 평가할 수 있다. 식 (10)을 w_l에 대한 함수 $\pi(w_l)$라 하면 추적자는 hybrid 추적방식 $\pi(w_l)$를 이용하여 최대한 빠른 시간 내에 도망자를 잡아 높은 점수를 얻기 위해 노력한다. 주어진 초기조건에서 기대값 $V^{\pi(w_l)}$은 다음과 같다.

$$V^{\pi(w_l)}=E[R|\pi(w_l)] \quad (15)$$

여기서 랜덤변수 R은 식 (16)과 같이 구할 수 있다.

$$R=\sum_{p_i=1}^{N_P}\sum_{t=0}^{t_f} r_{t,p_i} \tag{16}$$

r_{t,p_i}는 시간 t에서 추적자 p_i가 받는 보상으로 -1로 정하였다. 따라서 빠른 시간 내에 도망자를 잡을수록 높은 보상을 기대할 수 있다. 그리고 t_f는 추적-회피 게임이 끝나는 시간이다.

4.2 에피소드 매개변수 최적화 알고리즘

앞서 언급한 식 (15)의 기대값을 최대화 하는 가중치 w_l을 찾는 것이 목표고 기대값을 정확히 계산하는 것은 다루기 힘든 문제이므로 대신 N_{ep}번의 추적-회피 게임을 통해 얻은 전체 보상의 평균을 사용한다.

여기서 N_{ep}는 몬테카를로 형식의 시뮬레이션인 추적-회피 게임의 전체 에피소드의 수를 말한다. n_{ep}번째 추적-회피 게임에서 시간 $t=0$부터 $t=t_f$까지 보상 $r_{0,p_i}^{(n_{ep})}, r_{1,p_i}^{(n_{ep})}, \ldots, r_{t_f,p_i}^{(n_{ep})}$에 대해 식 (18)로부터 $R^{(n_{ep})}$를 얻을 수 있고, 이러한 과정을 $n_{ep}=1,\ldots,N_{ep}$까지 반복함으로써 식 (17)에 의해 전체보상 평균 R을 구할 수 있다. 전체보상 평균 R이 가중치 w_l에 대한 평가지표가 될 것이고, 다른 가중치 w_l'에 대한 전체보상 평균 R'을 같은 방식으로 구하고, 이 과정을 수렴할 때까지 반복하면 최적의 가중치 w_l^*을 구할 수 있다. 최적의 가중치 w_l^*의 적합도를 높이기 위해 전체 추적-회피 게임 에피소드의 수 N_{ep}는 충분히 크게 설정해야 하고 임의의 w_l에 대해 미리 정해진 같은 초기 조건 하에 최적화가 수행되어야 한다. 평균보상을 최대화 하는 방향으로 최적의 가중치를 찾기 위한 방법으로 황금분할법(Golden section search)[10]을 사용하였다. 자세한 내용은 Algorithm 1에서 확인 할 수 있고 Table 1은 에피소드 매개변수 최적화 알고리즘의 입력 변수들을 보여준다.

Table 1. Input parameters of the EPO algorithm

Parameter	Explanation	Value
q_1	current bound	0
q_2	center point	0.375
q_3	current bound	1
q_4	search point	0.625
ϵ	tolerance parameter	1.0e-004
ϕ	golden ratio	$\frac{1+\sqrt{5}}{2}$
N_{ep}	total number of episodes ($1 \le n_{ep} \le N_{ep}$)	1000

$$\mathrm{R}=\frac{1}{N_{ep}}\sum_{n_{ep}=1}^{N_{ep}} R^{(n_{ep})} \tag{17}$$

$$R^{(n_{ep})}=\sum_{p_i=1}^{N_P}\sum_{t=0}^{t_f} r_{t,p_i}^{(n_{ep})} \tag{18}$$

Algorithm 1.

Episodic parameter optimization algorithm

procedure EPISODIC PARAMETER OPTIMIZATION
Setup N_{ep} initial conditions of PEG
$n_{ep} \Leftarrow 1$
$\mathrm{R} \Leftarrow 0$
$w_l \Leftarrow q_2$
for $n_{ep} \le N_{ep}$ **do**
  Start from the n_{ep}-th initial conditions
  Run episode n_{ep} of PEG and update $R^{(n_{ep})}$
  $n_{ep} \Leftarrow n_{ep}+1$
end for
Compute R using (17)
$\mathrm{R}^{q_2} \Leftarrow \mathrm{R}$
while $|q_3-q_1| < \epsilon(|q_2|+|q_4|)$ **do**
  $n_{ep} \Leftarrow 1$
  $\mathrm{R} \Leftarrow 0$
  $w_l \Leftarrow q_4$
  for $n_{ep} \le N_{ep}$ **do**
    Start from the n_{ep}-th initial conditions
    Run episode n_{ep} of PEG and update $R^{(n_{ep})}$
    $n_{ep} \Leftarrow n_{ep}+1$
  end for
  Compute R using (17)
  $\mathrm{R}^{q_4} \Leftarrow \mathrm{R}$
  if $\mathrm{R}^{q_4} > \mathrm{R}^{q_2}$ **do**
    $\mathrm{R}^{q_2} \Leftarrow \mathrm{R}^{q_4}$
    if $q_3-q_2 > q_2-q_1$ **do**
      $q_1 \Leftarrow q_2;\ q_2 \Leftarrow q_4;\ q_3 \Leftarrow q_3;$
    else
      $q_1 \Leftarrow q_1;\ q_2 \Leftarrow q_4;\ q_3 \Leftarrow q_2;$
    end if
  else
    $\mathrm{R}^{q_2} \Leftarrow \mathrm{R}^{q_2}$
    if $q_3-q_2 > q_2-q_1$ **do**
      $q_1 \Leftarrow q_1;\ q_2 \Leftarrow q_2;\ q_3 \Leftarrow q_4;$
    else
      $q_1 \Leftarrow q_4;\ q_2 \Leftarrow q_2;\ q_3 \Leftarrow q_3;$
    end if
  end if
  if $q_3-q_2 > q_2-q_1$ **do**
    $q_4 \Leftarrow q_2+(2-\phi)(q_3-q_2)$
  else
    $q_4 \Leftarrow q_2-(2-\phi)(q_2-q_1)$
  end if
end while
return $w_l^* \Leftarrow \frac{q_3+q_1}{2}$
end procedure

V. 강화학습을 통한 최적정책

본 절에서는 4.2절의 에피소드 매개변수 최적화 알고리즘을 이용하여 추적-회피 게임 시나리오별 최적의 추적정책을 찾고자 한다.

5.1 최적화 시뮬레이션 설정

모든 최적화 시뮬레이션은 대해서 Table 2와 같이 설정된 상태에서 진행되었고, 각 셀의 크기가 10cm × 10cm 인 300cm × 300cm 의 공간상에서 도망자 수의 변화에 대한 최적 가중치의 경향성을 보기 위해 추적자의 수 N_P를 5기로 고정하였고, 도망자의 수 N_E를 1기에서 5기로 증가시켜가며 각 경우에 대해 가중치 최적화를 수행하였다. 추가적으로 추적-회피 게임이 이루어지는 각 셀의 크기가 10cm × 10cm 인 전체 공간의 크기를 200cm × 200cm, 300cm × 300cm, 400cm × 400cm 로 변화시켜감에 따라 추적자 5기와 도망자 1기, 추적자 5기와 도망자 5기의 각 경우에 대하여 최적화 시뮬레이션을 수행하였다. 위의 조건에 따른 각각의 가중치 w_l에 대한 추적-회피 게임의 전체 에피소드 수 N_{ep}는 1000번으로 설정하였다. 1000회의 에피소드별 추적자들과 도망자들의 위치, 방향각에 대한 초기 조건은 미리 정해져 있으므로 동일한 조건하에서 가중치 w_l의 최적화가 이루어진다.

5.2 최적화 시뮬레이션 결과

Figure 2는 추적자 5기에 대해 도망자의 수를 1기에서 5기로 증가시켰을 때 최적화 된 가중치 w_l^*의 값의 변화를 보여주고 있다. 도망자의 수가 1기인 경우 $w_l^* = 0.406061$의 값으로 선정되어 global-max의 영향이 local-max에 대한 영향보다 크게 작용하는 것을 알 수 있다. 도망자가 1기인 경우에는 추적자들이 한 곳으로 몰리더라도 단점으로 작용하지 않기 때문에 global-max의 영향이 local-max의 영향보다 크게 작용하는 것이 유리하다는 것을 알 수 있다. 반면 다수의 도망자의 경우 하나의 추적자가 도망자 한 기를 발견하였을 때 발견한 추적자 외의 멀리 떨어져 있는 추적자는 굳이 발견된 도망자를 잡기 위해 올 필요가 없고 근방에서 도망자를 탐색하는 것이 더 유리할 것이다. 결과적으로 Fig. 2에서 볼 수 있듯이 도망자의 수가 증가함에 따라 w_l^*의 값 역시 커지고 local-max가 global-max에 비해 더 큰 영향을 미치게 된다.

Table 2. Simulation parameters

Parameter	Value	Parameter	Value
Δt	0.25	K_p	1.0
A	200	p_{move}	0.75
B	150	$r_r(p_i, e_j)$	(15.6,31.2)

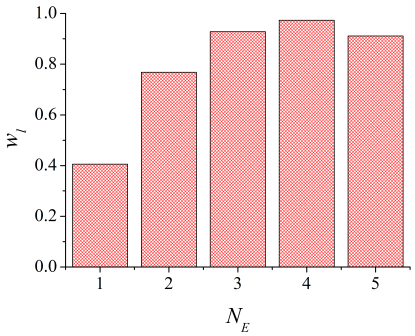

Fig. 2. The value of optimal weighting parameter w_l for the number of the evaders

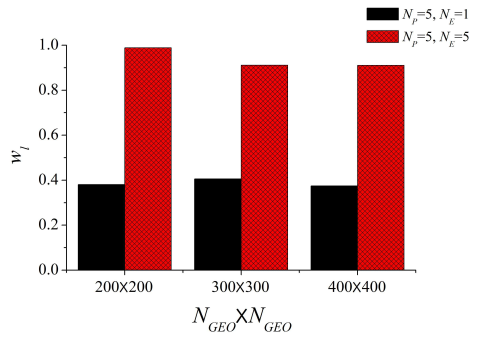

Fig. 3. The value of weighting parameter w_l for the size of the space and the number of the evaders

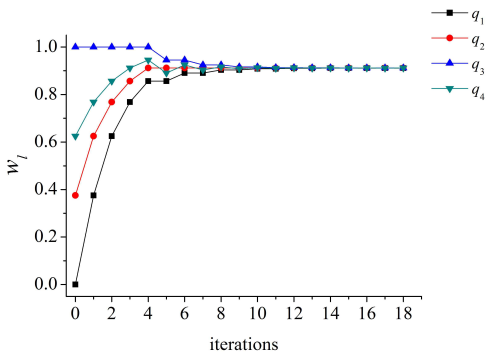

Fig. 4. Change in the value of the search points as the number of iterations increases ($N_{GEO} \times N_{GEO} = 300 \times 300$, $N_P = 5$, and $N_E = 5$)

Figure 3은 공간의 크기 변화에 대해 추적자가 5기 일 때 도망자 1기와 도망자 5기에 대해서 최적화 시뮬레이션을 수행한 결과를 보여준다. 도망자가 5기 일 때 공간의 크기가 커짐에 따라 w_l^*의 값은 약간 작아지긴 하지만 여전히 큰 값을 가져 local-max의 성향을 주는 것이 유리하다는 것을 알 수 있다. 그리고 도망자의 수가 1기일 경우에는 local-max와 global-max의 성향을 4:6의 비율로 설정하는 것이 추적성능을 높일 수 있을 것이다.

Figure 4는 공간 300㎝×300㎝에서 추적자 5기와 도망자 5기에 대한 최적화 시뮬레이션 반복 횟수가 증가함에 따른 탐색 점과 탐색 경계의 변화를 보여주고 있다.

VI. 결 론

본 논문에서는 확률기반 추적-회피 게임에서 global-max와 local-max 추적방식의 장점을 취한 hybrid 추적방식이 고려되었다. hybrid 추적방식의 local-max와 global-max의 성향을 조절하기 위한 가중치를 최적화하기 위해 강화학습을 기반으로 하고 있는 에피소드 매개변수 최적화 알고리즘을 적용하였다. 이 에피소드 매개변수 최적화 알고리즘을 이용하여 지능적인 도망자에 대해 추적자의 추적성능을 향상 시킬 수 있었고, 다수의 도망자에 대해 local-max에 대한 성향이 크게 나타난다는 것을 확인하였다. 또한, 추적-회피 게임이 이루어지는 공간의 크기 변화에 대해 각각 최적의 추적정책을 찾을 수 있었다. 본 논문에서 제안한 기법을 통해 추적-회피 게임 시나리오별 최적의 추적정책을 찾을 수 있을 것으로 기대되고 더 나아가 도망자가 추적자에 대응하여 좀 더 좋은 회피성능을 보여줄 수 있는 최적의 회피정책을 찾는 것이 앞으로의 과제다.

후 기

본 연구는 방위사업청 지정 국방무인화기술 특화연구센터 (UTRC) 및 교육과학기술부의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No.2011-0020423)

참고문헌

1) Isaacs, R., Differential *games: a mathematical theory with applications to warfare and pursuit, control and optimization*, Wiley, New York, 1965.

2) Vidal, R., Shakernia, O., Kim, J., Shim, D., and Sastry, S., "Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation," *IEEE Trans. on Robotics and Automation*, 2002, Vol. 42, pp. 662~669.

3) Schenato, L., Oh, S., and Sastry, S., "Swarm coordination for pursuit evasion games using sensor networks," *In Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 2493~2498.

4) Kwak, D. and Kim, J., "Probabilistic Pursuit-Evasion Game," *In Proceedings of KACC 2009*.

5) Kwak, D. and Kim, J., "Probabilistic Pursuit-Evasion Game," *In Proceedings of KSAS Fall 2009 Conference*, 2009, pp. 709~712.

6) Kwak, D. and Kim, J., "Probabilistic Pursuit-Evasion Game using Reinforcement Learning," *In Proceedings of KSAS Fall 2011 Conference*, 2011.

7) I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin, "Effective leadership and decision-making in animal groups on the move," Nature, 2005, vol. 433, no. 7025, pp. 513 - 516.

8) Khosla, P. and Volpe, R., "Superquadric artificial potentials for obstacle avoidance and approach," *In Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, 1988, pp. 1778~1784.

9) Sutton, R. S. and Barto, A. G., *Reinforcement learning: an introduction*, MIT Press, Cambridge, Mass., 1998.

10) Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical recipes in C: The art of scientific programming (2nd ed.)*, Cambridge: Cambridge University Press. 1992.