J. Korean Soc. Aeronaut. Space Sci. 50(4), 287-295(2022)
DOI:https://doi.org/10.5139/JKSAS.2022.50.4.287
ISSN 1225-1348(print), 2287-6871(online)

근사적 동적계획을 활용한 요격통제 및 동시교전 효과분석

이창석[1], 김주현[2], 최봉완[3], 김경택[4]

Approximate Dynamic Programming Based Interceptor Fire Control and Effectiveness Analysis for M-To-M Engagement

Changseok Lee[1], Ju-Hyun Kim[2], Bong Wan Choi[3] and Kyeongtaek Kim[4]
Department of Industrial Engineering, Hannam University, Daejeon, Republic of Korea[1,3,4]
Hanwha System, Seongnam, Republic of Korea[2]

ABSTRACT

As low altitude long-range artillery threat has been strengthened, the development of anti-artillery interception system to protect assets against its attacks will be kicked off. We view the defense of long-range artillery attacks as a typical dynamic weapon target assignment (DWTA) problem. DWTA is a sequential decision process in which decision making under future uncertain attacks affects the subsequent decision processes and its results. These are typical characteristics of Markov decision process (MDP) model. We formulate the problem as a MDP model to examine the assignment policy for the defender. The proximity of the capital of South Korea to North Korea border limits the computation time for its solution to a few second. Within the allowed time interval, it is impossible to compute the exact optimal solution. We apply approximate dynamic programming (ADP) approach to check if ADP approach solve the MDP model within processing time limit. We employ Shoot-Shoot-Look policy as a baseline strategy and compare it with ADP approach for three scenarios. Simulation results show that ADP approach provide better solution than the baseline strategy.

초 록

저고도 궤적의 장사정포 위협이 대두됨에 따라 이를 방어할 요격 시스템의 개발이 시작될 예정이다. 이러한 장사정포의 공격을 방어하는 문제는 전형적인 동적 무기 표적 할당 문제다. 동적 무기 표적 할당 문제에서는 한 시점에서의 의사결정 결과가 이후 시점의 의사결정 과정에 영향을 주며, 이는 마코브 의사결정 모형의 특징이기도 하다. 장사정포의 공격을 방어하기 위한 의사결정 과정에 허용되는 시간은 공격자와 방어자의 거리를 고려할 때 저고도 궤적의 동시 다발성 발사체에 대한 대응은 수 초 이내에 결정되어야 하나, 짧은 시간 내에 마코브 의사결정 과정으로 최적해를 구하는 것은 불가능하다. 본 논문에서는 장사정포 공격을 방어하는 동적 무기 표적 할당 문제를 마코브 의사결정 문제로 나타내고, 3가지 시나리오를 작성한 후 근사적 동적계획 방법을 적용하여 요격이 가능 시간 안에 해의 도출이 가능한지를 시뮬레이션을 통하여 확인하였다. 도출된 해의 품질을 검증하기 위하여 각 시나리오에 대하여 근사적 동적계획을 적용한 결과와 Shoot-Shoot-Look 방법을 적용한 결과를 비교하였다. 시뮬레이션 결과, 장사정포의 방어 시나리오에 대하여 근사적 동적계획의 결과가 Shoot-Shoot-Look 방법을 이용한 결과보다 우수함을 보였다.

† Received : February 22, 2022 Revised : March 20, 2022 Accepted : March 28, 2022
[1] Graduate Student, [2] Senior Engineer, [3,4] Professor
[4] Corresponding author, E-mail : kkim610@gmail.com, ORCID 0000-0002-5636-2487

Key Words : Long-Range Artillery(장사정포), Dynamic Weapon Target Assignment(동적 무기 표적 할당), Markov Decision Process(마코브 의사결정 과정), Approximate Dynamic Programming(근사적 동적계획), Effectiveness Analysis for Engagement(교전 효과분석)

Ⅰ. 서 론

최근 단시간에 동시 다발적으로 자산을 공격할 수 있는 저고도 궤적의 신형 방사포에 의한 북한 위협이 증대하고 있다[1,2]. 한국은 국가 중요시설뿐만 아니라 군사시설 등의 위협에 대비하기 위하여 장사정포 요격체계 구축을 추진하고 있다. 장사정포 요격체계는 여러 장소에 요격유도탄 포대를 기반으로 돔 형태의 방공망을 구축하는 개념으로 추진될 것으로 예상된다[3-6]. 다수의 장사정포로 아측 자산에 대한 공격이 이루어질 경우 방어자는 보유한 탐지/추적자산 및 요격체계를 활용하여 아측 피해를 최소화해야 할 것이다. 이러한 목적을 달성하기 위한 장사정포 요격체계를 구축하기 위해서는 요격포대의 효율적 배치뿐만 아니라[7] 방어자가 요격이 가능한 시간 안에 최대의 교전효과를 도출할 수 있는 교전계획의 수립이 매우 중요하다 할 수 있다.

그동안 무기 표적 할당(WTA, Weapon Target Assignment) 문제는 위협을 최소화하는 해법을 제시하기 위해 모형을 선형화하거나 유전자 알고리즘, 타부(tabu) 탐색, PSO(Particle Swarm Optimization) 등을 활용하여 근사해를 구하는 연구들이 주를 이루었다[1,8-10]. 그러나 위협수준 및 요격체계 활용에 있어서 현황의 복잡성 때문에 비교적 단순하게 평가하거나 정적(static) 무기 표적 할당 문제의 해를 구하였다[11,12]. 탄도미사일에 대한 방어포대 할당 및 교전효과에 대해 다수의 연구가 진행되었지만 저고도 궤적의 동시 다발성 장사정포 방어체계에 대한 연구는 초기단계이다[13-15].

최근 연구결과에서 탄도미사일의 비행궤적을 정확하게 예측하는 것이 요격유도탄 할당에 매우 중요하다고 판단하였으며[13,14], 이를 바탕으로 위협발사체의 궤적을 분석하여 규칙기반의 효과적인 요격유도탄 할당 방법론을 제안하였다[15]. 또한, 교전효과 증대를 위해 다수의 위협에 대한 실시간 교전효과를 판단해보기 위한 요격 성공인자에 대한 연구[16]와 실시간 무장할당을 위한 강화학습과 평균 필드 이론을 적용한 연구가 진행되었다[17]. 국외에서도 탄도미사일 위협에 대한 요격유도탄의 할당문제가 마코브 의사결정과정(MDP, Markov Decision Process)을 따르지만 최적의 할당을 제시하기 어려우므로 근사해를 찾는 방법을 제안하고 있다[18,19].

최근의 저고도 궤적의 동시 다발성 장사정포의 위협상황은 방어모형의 복잡성과 해법을 구하는 데 필요한 시간의 제약 때문에 기존 방법론으로 최적해를 구하는 것은 거의 불가능에 가깝다고 판단된다.

따라서 본 연구에서는 국내·외적으로 연구되고 있는 탄도미사일 무장할당 이론에 착안, 저고도 궤적의 동시 다발성 장사정포의 위협 대비 효과적인 요격통제와 교전 결과를 도출하기 위한 ADP 방법론 적용과 검증을 아래와 같이 3단계로 추진하였다.

첫째, 저고도 궤적의 동시 다발성 장사정포의 위협을 MDP로 모형화한 후 최적해 도출의 시간 복잡도를 살펴보았다.

둘째, MDP 모형의 복잡성과 해법의 시간 제약을 극복할 수 있는 방법론으로 강화학습기반의 ADP 알고리즘을 적용하였다.

셋째, 장사정포 공격에 대한 교전상황을 반영한 3가지 시나리오에 대한 시뮬레이션을 수행하여 본 연구에서 적용한 ADP 알고리즘의 우수함을 확인하였다.

Ⅱ. 본 론

2.1 문제의 정의/가정사항

현재 동시 다발성 장사정포의 저고도 궤적은 적의 공격 이전에 식별할 수 없고, 아측의 대응시간이 짧아 Shoot-Look (요격 유도탄 1회 발사) 또는 Shoot-Shoot-Look (요격 유도탄 2회 발사) 방법을 적용할 수밖에 없다. 본 논문에서 논의를 진행하기 위하여 유사한 연구[18,19]와 ADP 관련 서적[20]에서 사용한 기호를 참고하여 아래와 같이 문제를 정의한다.

$\mathrm{T}=\{1,2,\cdots,T\}, T\le\infty$를 의사결정 시점(epoch)의 집합이라고 하고, t를 임의의 의사결정 시점이라 한다. 적의 공격으로부터 방어해야 할 다수의 자산이 있으며, t시점에서의 자산 i의 상태는 $A_t=(A_{ti})_{i\in\mathrm{A}}\equiv(A_{t1},A_{t2},\cdots,A_{t|\mathrm{A}|})$로 나타낸다. 여기에서 $\mathrm{A}=\{1,2,\cdots,|\mathrm{A}|\}$는 모든 자산의 집합이다. 아울러 $A_{ti}\in\{0,0.05,0.1,...,0.95,1\}$이며 $A_{ti}=1$은 장사정포에 의한 피해가 전혀 없는 상태를, $A_{ti}=0$은 자산의 가치가 전혀 없는 상태를 나타낸다.

각 자산의 가치가 v_i일 때 시간 t에서 시간 $t+1$ 동안 전체 자산 가치의 감소는 $\sum_{i\in\mathrm{A}}v_i(A_{ti}-A_{t+1,i})$로 나타낸다.

아측은 자산을 방어할 요격유도탄 포대를 보유하고 있으며, 자산 i에 위치한 포대의 t시점에서의 탄약 재고는 $R_t=(R_{ti})_{i\in\mathrm{A}}\equiv(R_{t1},R_{t2},\cdots,R_{t|\mathrm{A}|})$로 나타낸다.

t시점에서의 자산 i를 향해 발사된 적 장사정포의 공격 벡터는 $\hat{M}_t=(\hat{\mathrm{M}}_{ti})_{i\in\mathrm{A}}\equiv(\hat{\mathrm{M}}_{t1},\hat{\mathrm{M}}_{t2},\cdots,\hat{\mathrm{M}}_{t|\mathrm{A}|})$로 나타낸다. 여기에서 $\hat{\mathrm{M}}_t$는 t시점이 되어야만 알 수 있

으며, hat(ˆ) 기호는 외생(exogenuous) 정보임을 나타내기 위하여 사용한다.

포대의 초기 요격유도탄 재고는 분산된 기지나 일정 지역 내에 독립된 포대의 개념으로 배치된 것을 의미한다. 요격유도탄은 장사정포의 공격을 방어하기 위해 각 기지/포대별로 일정 방어 반경을 갖고 있다. 적 장사정포 공격에 대한 효과적인 요격에 실패하면 자산은 장사정포의 파괴력 및 자산의 취약성을 고려하여 축차적으로 파괴된다. 적의 장사정포 공격은 아측 자산을 대상으로 포대별 보유량을 동시에 축차적으로 발사하며, 목표는 장사정포 포대별 하나의 자산을 지정한다. 동시에 두 개 이상의 장사정포가 하나의 아측 자산을 공격할 수 있다. 적의 장사정포 공격이 시작된 후에 어느 자산으로 궤적이 발생하는지 식별이 가능하고, 이에 따라 아측 포대의 요격유도탄을 할당하고 교전을 수행할 수 있다. 이때 교전효과를 기반으로 자산피해가 최소화되는 것을 목표로 한다. 위에서 설명한 문제는 동적 무기 표적 할당(DWTA, Dynamic Weapon Target Assignment) 문제로 정의할 수 있으며, 이는 MDP 모형으로 표현할 수 있다.

2.2 MDP 모형

MDP 모형은 환경의 상태를 나타내는 상태(state), 취할 수 있는 행동(action), 상태 S에서 행동 a를 취했을 때 나타나는 전이 확률(transfer probability) 또는 전이 함수(transfer function), 상태 S에서 행동 a를 취했을 때 받는 보상(reward), 현재 얻게 되는 보상이 미래에 얻는 보상보다 얼마나 중요한지를 나타내는 할인인자(discount factor)의 5가지 항목에 의해 정의된다.

동적 무기 표적 할당 문제에서는 전장 상황의 변화에 따라 최적의 순차적 의사결정을 해야 하는데, 이를 위하여 할당 문제를 앞 절에서 사용한 기호를 사용하여 MDP로 모형화한다. MDP 모형에서 목적함수는 자산 가치 손실의 최소화(또는 보상의 최대화)이며, 앞에서 언급한 5가지 항목을 사용하여 의사결정을 한다.

상태는 $S_t = (A_t, R_t, \hat{M}_t) \in \mathrm{S}$로 나타내며, 여기에서 S는 모든 가능한 상태의 전체 집합이다.

$x_{tij} \in \mathbb{N}^0$를 t시점에서 요격유도탄 포대 i에서 장사정포 j를 향해 발사된 요격유도탄의 수라 한다. 요격유도탄의 수는 포대의 요격유도탄 재고와 연속 발사 제약 등에 따라 제한을 받는다.

$x_t = (x_{tij})_{i \in \mathrm{A}, j \in \hat{\mathbb{M}}_{ti}}$를 벡터 형태로 표현된 요격유도탄이라 하면 모든 가능한 요격유도탄 벡터는 식 (1)과 같다.

$$\chi_{S_t} = \left\{ x_t : \sum_{j \in \hat{M}_{ti}^A} x_{tij} \le \min\left(R_{ti}, x_i^{\max}\right), \forall i \in A \right\} \tag{1}$$

$x_i^{\max}$는 연속 발사할 수 있는 최대 요격유도탄의 수를 나타낸다.

상태 전이함수는 $S_{t+1} = S^M(S_t, x_t, W_{t+1})$로 정의하며, 여기에서 $W_{t+1} = (\hat{A}_{t+1}, \hat{M}_{t+1})$는 확률 변수로 $t+1$시점에서 알게 되는 정보를 나타낸다. 여기에서 $\hat{A}_{t+1}$은 t시점에서 장사정포 공격 $\hat{M}_t$와 요격유도탄 벡터 x_t에 의해 결정되는 확률 변수다.

요격유도탄 재고 전이 함수는 식 (2)와 같이 t에서의 재고에서 t에서의 발사된 요격유도탄의 수를 빼면 된다.

$$R_{t+1,i} = R_{ti} - \sum_{j \in \hat{M}_{ti}^A} x_{tij}, \forall i \in A \tag{2}$$

t시점에서 요격유도탄 벡터의 결정에도 불구하고, 요격유도탄이 장사정포 탄을 피격했는지는 확정되지 않고, 확률적으로 표현된다. 따라서 확률적으로 표현되는 즉각적인 자산 가치의 손실(Cost)은 식 (3)과 같이 표현된다.

$$\hat{C}(S_t, x_t, \hat{A}_{t+1}) = \sum_{i \in \mathrm{A}} v_i (A_{ti} - \hat{A}_{t+1,i}) \tag{3}$$

여기에서 v_i는 자산 i의 가치다. 기댓값을 사용하여 비용 함수를 현재의 상태와 현재의 요격유도탄 벡터의 함수로 표현하면 식 (4)와 같다.

$$C(S_t, x_t) = \mathbb{E}\left\{ \sum_{i \in \mathrm{A}} v_i (A_{ti} - \hat{A}_{t+1,i}) \mid S_t, x_t \right\} \tag{4}$$

MDP 모형에서는 앞으로 입게 되는 피해의 총합을 최소화할 수 있는 행동(즉, 요격 벡터의 결정)들을 알고리즘을 통해 계산한다. 각 상태별로 결정되는 행동을 정책(policy)이라 부른다. $X^\pi(S_t)$를 요격유도탄 벡터를 결정하는 함수(즉, 정책)라 하면 목적함수는 식 (5)와 같이 표현할 수 있다.

$$\min_{\pi \in \Pi} \mathbb{E}^\pi \left\{ \mathbb{E}^T \left\{ \sum_{t=1}^{T} C(S_t, X^\pi(S_t)) \right\} \right\} \tag{5}$$

여기에서 $\mathbb{E}^\pi$는 정책 π의 기댓값을 나타내며, $\mathbb{E}^T$는 적군의 공격 기간(time horizon) T동안의 기댓값을 의미한다. 적군의 공격 기간이 무한대이고 요격유도탄의 재고가 유한하다면 자산은 결국 모두 파괴될 것이고, 어떤 정책에 대해서도 목적함수는 동일한 값을 갖게 된다. 따라서 자산이 나중에 파괴될수록 피해가 작아지도록 할인인자(discount factor)를 사용하여 표현하면 식 (6)과 같이 표현할 수 있다.

$$\min_{\pi \in \Pi} \mathbb{E}^\pi \left\{ \sum_{t=1}^{\infty} \gamma^{t-1} C(S_t, X^\pi(S_t)) \right\} \tag{6}$$

S_t의 가치를 $J(S_t)$라 하면 최적의 정책은 다음의 Bellman 방정식의 해를 구하여 얻을 수 있으며,

$$J(S_t) = \min_{x_t \in \chi_{S_t}} (C(S_t, x_t) + \gamma \mathbb{E}\{J(S_{t+1}) \mid S_t, x_t\}) \quad (7)$$

그 해는 식 (8)과 같다.

$$X^{\pi}(S_t) = \arg\min_{x_t \in \chi_{S_t}} (C(S_t, x_t) + \gamma \mathbb{E}\ \{J(S_{t+1}) \mid S_t, x_t\}) \quad (8)$$

2.3 ADP 모형

MDP 모형에서 Bellman 방정식을 이용하여 DWTA 문제에 대한 정확한 최적해를 구할 수 있지만, 실제 문제를 MDP로 모형화하여 Bellman 방정식으로 최적해를 구할 때 두 가지 문제에 직면한다[20].

첫째 문제는 MDP로 모형화하면 상태의 크기가 너무 커서 소규모의 문제에서만 실제적인 계산이 가능하다는 사실에 있다[20]. 예를 들어 방어할 자산이 5곳이고 한번 공격마다 적의 장사정포가 12발 단위로 동시에 최대 3번 발사할 수 있다고 가정하자. 요격유도탄 포대가 5곳 있으며, 한 포대가 12발씩 10번 발사가 가능한 요격유도탄을 가지고 있고 장사정포 1발은 자산의 최초 가치를 0.05배씩 감소시킨다고 가정한다. 이 경우 한 시점에서의 상태(S_t)의 수는 최대 3.6×10^{13}가 된다. 이는 문제의 규모가 조금만 커져도 Bellman 방정식을 이용하여 정확한 해를 구하는 방법은 실제상황에서 사용하기 어렵다는 사실을 보여준다. 이러한 사실은 MDP 모형의 한계점으로 자주 언급되는 사항으로 실제 연산을 통하여 MDP의 모든 요소(S_t, A_t, R_t, π)의 값을 구하는 것은 불가능하다.

ADP에서는 상태의 수를 줄이는 방법으로 의사결정 직후(post-decision) 상태라는 개념[20]을 DWTA 문제에 적용해보면, 의사결정 직후 상태는 요격 벡터를 결정한 직후로 아직 새로운 정보 W_{t+1}이 도착하기 직전의 상태를 나타낸다. 의사결정 직후 상태를 S_x^t라 하면 상태 전이 함수 $S_{t+1} = S^M(S_t, x_t, W_{t+1})$는 식 (9)와 같이 두 단계로 나눌 수 있다.

$$\begin{aligned} S_t^x &= S^{M,x}(S_t, x_t) \\ S_{t+1} &= S^{M,W}(S_t^x, W_{t+1}) \end{aligned} \quad (9)$$

이때 $S_x^t = (A_x^t, R_x^t)$로 나타낸다. 따라서 MDP 모형에 비하여 상태의 크기가 $1/\left|\widehat{M}_t\right|$배로 줄어든다. 앞의 상태($S_t$)의 크기 계산 사례에 적용하면 ADP 모형에서 상태의 크기는 MDP 모형의 경우보다 1/56 배가 된다. ADP 모형에서 상태의 크기가 줄어들었지만 여전히 실제 문제에 대한 적용이 불가능하다.

둘째 문제는 MDP 모형에서는 어떤 상태의 가치를 계산하기 위해서는 그 상태로부터 이후 모든 시간에 걸쳐 직간접적으로 전이될 수 있는 모든 상태의 가치를 알고 있어야 한다는 사실에 있다. 그러므로 매우 간단한 경우를 제외한 대부분의 실제적인 DWTA 문제에 있어서 그 가치를 정확하게 계산하는 것은 불가능하다. 이에 비하여 ADP에서는 근사적 가치함수를 사용하여 계산하며, 근사적 가치함수를 추정할 때 입력으로 사용하는 상태의 수는 사용자가 파라미터로 조절할 수 있다.

의사결정 직후 상태 S_x^t의 가치를 $J^x(S_t^x)$라 하면 $J(S_t)$와 $J^x(S_t^x)$의 관계는 식 (10), (11)과 같다[20,21].

$$J(S_t) = \min_{x_t \in \chi_{S_t}} (C(S_t, x_t) + \gamma J^x(S_t^x)) \quad (10)$$

$$J^x(S_t^x) = \mathbb{E}\{J(S_t) \mid S_t^x\} \quad (11)$$

식 (1)을 $J^x(S_{t-1}^x)$에 대입하면 의사결정 직후 상태에 대한 Bellman 방정식을 식 (12)와 같이 얻는다.

$$J^x(S_{t-1}^x) = \mathbb{E}\left\{\min_{x_t \in \chi_{S_t}} (C(S_t, x_t) + \gamma J^x(S_t^x)) \mid S_{t-1}^x\right\} \quad (12)$$

이 식에서 J를 선형 기저 함수(basis function)에 의해 결정된다고 가정하고 의사결정 직후 상태에 대한 가치를 근사적으로 추정하자. 이렇게 함으로써 정확한 값을 추정하지 못하는 단점은 있지만, 매우 빠른 계산시간 안에 DWTA 문제에 대한 근사해를 구할 수 있다. 가치의 추정에 사용되는 기저 함수를 $\phi_f(S_t^x)$라 하면 가치 추정치는 식 (13)과 같이 기저 함수들의 선형 회귀 직선으로 나타낼 수 있다.

$$\bar{J}_t^x(S_t^x) = \sum_{f \in \mathrm{F}} \theta_f \phi_f(S_t^x) \quad (13)$$

여기에서 θ_f는 파라미터 벡터를 나타낸다. 식(13)의 Bellman 방정식에서 $J^x(S_t^x)$ 대신에 $\bar{J}^x(S_t^x)$을 사용하여 다시 쓰면 식 (14)와 같다.

$$\bar{J}^x(S_{t-1}^x) = \mathbb{E}\left\{\min_{x_t \in \chi_{S_t}} (C(S_t, x_t) + \gamma \sum_{f \in \mathrm{F}} \theta_f \phi_f(S_t^x) \mid S_{t-1}^x\right\} \quad (14)$$

강화학습에서 정책을 개선하기 위하여 사용하는 정책 반복(Policy Iteration) 알고리즘은 정책 평가(Policy Evaluation)와 정책 개선(Policy Improvement)의 두 단계를 교차해가며 반복적으로 실행한다. 본 논문에서는 상태의 가치를 계산하는 정책 평가 단계에서 위 식 (14)에서 제시한 근사식을 사용하여 가치를 추정한다. 개선된 정책을 만드는 정책 개선 단계에서는 현재의 가치함수 근사치를 이용하여 새로운 정책을 만들어낸다.

시간차(Temporal Difference) 학습이란 가치함수를 업데이트할 때 공격 기간이 끝날 때까지 기다리지 않고, 각 의사결정 시점마다 업데이트하는 방법이다.

Initialize θ^0
set α
for n = 1 **do to** N (**Policy Improvement Loop**)
  for k = 1 **do to** K (**Policy Evaluation Loop**)
    Generate a random post-decision state $S^x_{t-1,k}$
    Record $\phi(S^x_{t-1,k})$
    Sample W_{t+1}
    Simulate next state $S_{t,k}$
    Determine decision $x_{t,k} = X^\pi(S_{t,k} \mid \theta^{n-1})$
    Record $C(S_{t,k}, x_{t,k})$
    Compute post-decision state $S^x_{t,k}$
    Record $\phi(S^x_{t,k})$
  end for
  Update θ^n :
  $\hat{\theta} = [(\Phi_{t-1} - \gamma\Phi_t)^T(\Phi_{t-1} - \gamma\Phi_t)]^{-1}(\Phi_{t-1} - \gamma\Phi_t)^T C_t$
  $\alpha_n = \dfrac{a}{a+n-1}$
  $\theta^n = \alpha_n\hat{\theta} + (1-\alpha_n)\theta^{n-1}$
end for
Return $X^\pi(S_t \mid \theta^N)$
End

Fig. 1. ADP Algorithm with LSTD [18,19,22]

시간차 학습은 여러 변형이 있는데, 본 논문에서 이용하는 Fig. 1의 ADP 알고리즘에서는 최소 자승 시간차(Least Square Time Difference, LSTD)를 사용한다. ADP 알고리즘은 N번의 정책개선 루프와 각 정책개선 루프 안에 K번의 정책 평가로 이루어져 있다. 먼저 θ^0를 초기화한 후에 K번의 정책평가 동안 기저함수와 Cost를 기록하며, 이를 마치면 정책 평가의 루프가 끝난다. 이들 배치(batch) 데이터를 이용하여 최소 자승 시간차를 이용하여 업데이트된 θ^1를 계산함으로써 정책 개선 루프가 끝난다. θ^1을 이용하여 정책평가 루프를 실행한 후 θ를 업데이트한다. 이런 식으로 θ^N을 구한 다음 θ^N을 이용하여 $X^\pi(S_t \mid \theta^N)$을 계산하면 알고리즘을 마치게 된다. θ를 업데이트하기 위한 수식에서 사용된 기저함수 행렬과 Cost 벡터는 식 (15)와 같다.

$$\Phi_{t-1} \triangleq \begin{bmatrix} \phi(S^x_{t-1,1})^T \\ \vdots \\ \phi(S^x_{t-1,K})^T \end{bmatrix}, \Phi_t \triangleq \begin{bmatrix} \phi(S^x_{t,1})^T \\ \vdots \\ \phi(S^x_{t,K})^T \end{bmatrix}, C_t \triangleq \begin{bmatrix} C(S_{t,1}, x_t) \\ \vdots \\ C(S_{t,K}, x_t) \end{bmatrix} \quad (15)$$

Figure 1의 알고리즘은 N*K개의 S^x_{t-1}와 이로부터 전이되는 S^x_t의 데이터만 가지고 가치함수 방정식의 계수를 추정하는 방법을 보여준다. 도출된 가치함수 추정 방정식은 모든 S^x_t의 가치를 추정하는 데 사용함으로써 앞에서 언급한 두 가지 문제를 모두 해결하고 짧은 시간 안에 근사해를 제시한다. 근사해의 품질은 기저함수의 채택에 영향을 받으므로 주어진 문제마다 검증해야 한다.

2.4 시뮬레이션 시나리오

적·아 배치는 Fig. 2에 요약되어 있다. 적의 장사정포 공격은 한 번의 공격에 최대 3개의 장소에서 공격할 수 있으며, 한 장소의 한 번의 장사정포 공격은 동일한 자산에 12발을 발사한다. 아측은 3개의 자산($i=1,2,3$)을 방어해야 하며, 각 자산의 가치는 (1, 10, 5)로 가정한다. 아측 요격유도탄 포대는 2개의 장소에 배치되어($j=1,2$), 포대 1은 자산 1, 2를 방어하고 포대 2는 자산 2, 3을 방어한다.

요격유도탄 포대는 한 번의 방어에 최대 4회의 요격유도탄을 사용할 수 있으며, 각 유도탄포대는 적 장사정포 공격에 대하여 최대 2회 방어한다고 가정한다. 1회의 요격유도탄은 적 장사정포 1회(12발)에 대하여 1회(12발) 발사한다. 요격유도탄의 명중률은 90%이며, 장사정포 1발이 자산에 명중하면 자산의 가치는 최초 가치의 5%씩 차감되는 것으로 가정한다.

적은 동일한 확률로 1, 2, 3의 장사정포를 사용하며, 아측 자산의 가치와 피해 상황을 알고 있다고 가정한다. 또한 아측 자산의 잔존가치에 따른 다항분포에 의거하여 자산을 선택하여 공격한다. 예를 들어 자산의 잔존가치가 (1, 10, 5)이면 (1/16, 10/16, 5/16)의 매개변수를 갖는 다항분포에 의한 자산을 선택하여 공격하며, 자산의 잔존가치가 (0, 6, 2)이면 (0/8, 6/8, 2/8)의 매개변수를 갖는 다항분포에 의하여 공격할 자산을 선택한다.

S^x_t의 가치를 근사적으로 추정할 때 사용하는 기저함수로 $\phi_0(S^x_t)=1$, $\phi_1(S^x_t)=A_t$, $\phi_2(S^x_t)=R^x_t$, $\phi_3(S^x_t)=A^x_t$로 하였으며, LSTD에서 θ를 업데이트할 때 사용

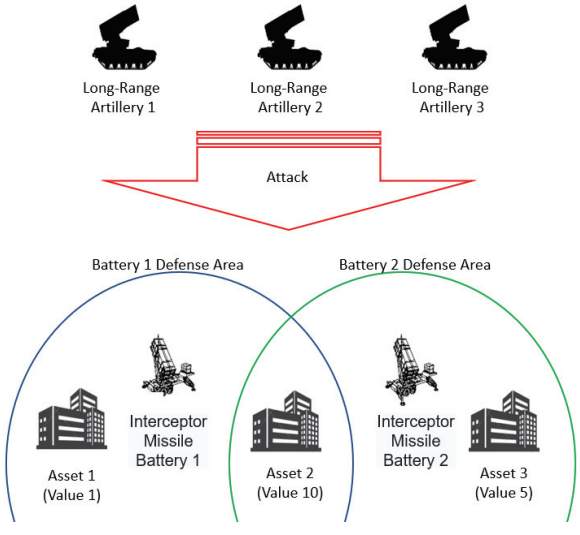

Fig. 2. Scenario Layout

하는 α는 1로, 할인인자 γ는 0.8로 고정하였다. 알고리즘의 정책개선 루프는 15번 실행하고, 정책평가 루프는 50번 실행하였다.

ADP 알고리즘의 결과와 비교할 대상으로 Shoot-Shoot-Look (SSL) 정책을 선택하였으며, SSL 정책은 장사정포 공격 1회에 대하여 요격유도탄 2회 발사하며, 자산 2가 공격받을 경우 요격유도탄 포대 1이 우선 대응하도록 하였다.

요격 유도탄 재고를 변경해가며 3개의 시나리오를 만들어 시뮬레이션을 실행하였다. 시나리오 1에서 R_t =(2,2), 시나리오 2에서 R_t=(4,4), 시나리오 3에서 R_t =(12,12)로 설정하였다. 모든 경우 A_t=(1,1,1)로 하였다. ADP 알고리즘으로 시뮬레이션을 실행하면 1초 이내에 결과를 얻었다. 이러한 빠른 실행결과는 LSTD 알고리즘의 시간 복잡도가 각 t에 대하여 $O(|\phi(S_t^x)|^3)$ 이기 때문이다[23].

2.5 시뮬레이션 결과 및 효과분석

2.5.1 시나리오 1: $R_t = (2,2)$

방어자가 2개의 포대에서 각 2회(24발)의 재고를 가질 때 공격자가 아측 3개 자산에 대하여 공격하는 모든 경우에 대하여 아측의 대응과 그에 따른 손실 가치를 Table 1에 나타내었다.

장사정포 공격에 대하여 ADP 알고리즘을 적용하는 경우 SSL을 적용한 대응 배치의 자산 손실에 비하여 16.85% 감소한 결과를 나타냈다. ADP는 자산 가치에 따라 선택적으로 대응을 전혀 하지 않거나 1회만 대응하는 패턴을 보인다. 이는 요격 유도탄의 재고가 충분하지 않은 상태를 반영하는 것으로 보인다. 예를 들면 자산 가치가 적은 자산 1에 대한 장사정포의 공격에 대해서는 요격유도탄의 재고가 있음에도 공격 벡터의 구성에 관계없이 ADP 알고리즘은 이에 대한 방어를 전혀 하지 않는다.

2.5.2 시나리오 2: $R_t = (4,4)$

Table 2는 방어자가 2개의 포대에서 각 4회(48발)의 요격 유도탄 재고가 있는 경우에 장사정포의 공격 벡터에 따른 의사결정 결과를 보여준다.

ADP 알고리즘을 사용하는 경우 SSL을 적용한 대응 배치의 자산 손실에 비하여 18.20% 감소한 결과를 나타냈다. ADP는 시나리오 1과 매우 유사하게 자산 가치에 따라 선택적으로 반응을 하지 않거나 1회만 반응하는 패턴을 보인다. 여전히 요격 유도탄의

Table 1. Policy Comparison for Scenario 1, $R_t = (2,2)$

Attack Probability ($\mathbb{P}(\widehat{M}_t)$)	Attack Vector ($\widehat{M}_t$)			ADP Policy ($X^\pi(S_t \mid \theta^N)$)						ADP Value* ($J(S_t \mid \theta^N)$) ①	SSL Value* ($J(S_t)$) ②	Gap* (%) (②-①)/②	SSL Policy ($X^\pi(S_t)$)					
				Battery 1			Battery 2						Battery 1			Battery 2		
0.1042	0	0	1	0	0	0	1	0	0	9.39	11.21	16.24	0	0	0	2	0	0
0.0326	0	0	2	0	0	0	1	1	0	10.61	12.69	16.39	0	0	0	2	0	0
0.0102	0	0	3	0	0	0	0	0	0	11.35	13.13	13.56	0	0	0	2	0	0
0.2083	0	1	0	1	0	0	0	0	0	9.52	11.22	15.15	2	0	0	0	0	0
0.1302	0	1	1	1	0	0	0	1	0	10.56	12.53	15.72	2	0	0	0	2	0
0.0610	0	1	2	1	0	0	0	1	1	11.71	14.05	16.65	2	0	0	0	2	0
0.0302	0	2	0	1	1	0	0	0	0	10.61	12.56	15.53	2	0	0	0	2	0
0.1221	0	2	1	1	1	0	0	0	1	11.61	14.08	17.54	2	0	0	0	2	0
0.0814	0	3	0	1	1	0	0	0	1	11.68	15.64	25.32	0	2	0	0	0	2
0.0208	1	0	0	0	0	0	0	0	0	9.17	11.17	17.91	2	0	0	0	0	0
0.0130	1	0	1	0	0	0	0	1	0	10.09	12.49	19.22	2	0	0	0	2	0
0.0061	1	0	2	0	0	0	0	1	1	11.25	14.00	19.64	2	0	0	0	2	0
0.0260	1	1	0	0	1	0	0	0	0	10.21	11.43	10.67	0	2	0	0	0	0
0.0244	1	1	1	0	1	0	0	0	1	11.21	12.74	12.01	0	2	0	0	0	2
0.0244	1	2	0	0	1	1	0	0	0	11.27	12.76	11.68	0	2	0	0	0	2
0.0013	2	0	0	0	0	0	0	0	0	9.17	11.38	19.42	2	0	0	0	0	0
0.0012	2	0	1	0	0	0	0	0	1	10.09	12.69	20.49	2	0	0	0	0	2
0.0024	2	1	0	0	0	1	0	0	0	10.22	11.33	9.80	0	0	2	0	0	0
0.0001	3	0	0	0	0	0	0	0	0	9.17	11.27	18.63	2	0	0	0	0	0
									$\mathbb{E}[\star]$=	9.45	11.37	16.85						

Table 2. Policy Comparison for Scenario 2, $R_t = (4,4)$

Attack Probability ($\mathbb{P}(\widehat{M}_t)$)	Attack Vector ($\widehat{M}_t$)			ADP Policy ($X^{\pi}(S_t \mid \theta^N)$) Battery 1			Battery 2			ADP Value* ($J(S_t \mid \theta^N)$) ①	SSL Value* ($J(S_t)$) ②	Gap* (%) (②-①)/②	SSL Policy ($X^{\pi}(S_t)$) Battery 1			Battery 2		
0.1042	0	0	1	0	0	0	1	0	0	7.43	9.09	18.26	0	0	0	2	0	0
0.0326	0	0	2	0	0	0	1	1	0	8.15	10.23	20.33	0	0	0	2	2	0
0.0102	0	0	3	0	0	0	0	0	0	10.32	11.83	12.76	0	0	0	2	2	0
0.2083	0	1	0	1	0	0	0	0	0	7.70	9.04	14.82	2	0	0	0	0	0
0.1302	0	1	1	1	0	0	0	1	0	8.38	11.18	25.04	2	0	0	0	2	0
0.0610	0	1	2	1	0	0	0	1	1	9.09	11.29	19.49	2	0	0	0	2	2
0.0302	0	2	0	1	1	0	0	0	0	8.58	10.17	15.63	2	2	0	0	0	0
0.1221	0	2	1	1	1	0	0	0	1	9.27	11.30	17.96	2	2	0	0	0	2
0.0814	0	3	0	1	1	1	0	0	0	9.41	11.33	16.95	2	2	0	0	0	2
0.0208	1	0	0	0	0	0	0	0	0	7.57	8.99	15.80	2	0	0	0	0	0
0.0130	1	0	1	0	0	0	0	1	0	8.22	10.13	18.85	2	0	0	0	2	0
0.0061	1	0	2	0	0	0	0	1	1	8.91	11.24	20.73	2	0	0	0	2	2
0.0260	1	1	0	0	1	0	0	0	0	8.49	10.13	16.19	2	2	0	0	0	0
0.0244	1	1	1	0	1	0	0	0	1	9.13	11.25	18.84	2	2	0	0	0	2
0.0244	1	2	0	0	1	1	0	0	0	9.34	10.41	10.28	0	2	2	0	0	0
0.0013	2	0	0	0	0	0	0	0	0	7.57	10.08	24.90	2	2	0	0	0	0
0.0012	2	0	1	0	0	0	0	0	1	8.23	11.20	26.52	2	2	0	0	0	2
0.0024	2	1	0	0	0	1	0	0	0	8.49	10.36	18.05	2	0	2	0	0	0
0.0001	3	0	0	0	0	0	0	0	0	7.57	10.31	26.58	2	2	0	0	0	0
									$\mathbb{E}[\star]=$	7.58	9.26	18.20						

Table 3. Policy Comparison for Scenario 3, $R_t = (12,12)$

Attack Probability ($\mathbb{P}(\widehat{M}_t)$)	Attack Vector ($\widehat{M}_t$)			ADP Policy ($X^{\pi}(S_t \mid \theta^N)$) Battery 1			Battery 2			ADP Value* ($J(S_t \mid \theta^N)$) ①	SSL Value* ($J(S_t)$) ②	Gap* (%) (②-①)/②	SSL Policy ($X^{\pi}(S_t)$) Battery 1			Battery 2		
0.1042	0	0	1	0	0	0	2	0	0	3.63	4.04	10.15	0	0	0	2	0	0
0.0326	0	0	2	0	0	0	2	2	0	4.08	4.50	9.33	0	0	0	2	2	0
0.0102	0	0	3	0	0	0	2	1	1	4.78	6.89	30.62	0	0	0	2	2	0
0.2083	0	1	0	2	0	0	0	0	0	3.62	4.10	11.71	2	0	0	0	0	0
0.1302	0	1	1	2	0	0	0	2	0	4.05	4.51	10.20	2	0	0	0	2	0
0.0610	0	1	2	2	0	0	0	2	2	4.53	4.99	9.22	2	0	0	0	2	2
0.0302	0	2	0	2	2	0	0	0	0	4.05	4.6	11.96	2	2	0	0	0	0
0.1221	0	2	1	2	2	0	0	0	2	4.51	5.04	10.52	2	2	0	0	0	2
0.0814	0	3	0	2	2	0	0	0	2	4.55	5.07	10.26	2	2	0	0	0	2
0.0208	1	0	0	1	0	0	0	0	0	3.47	4.05	14.32	2	0	0	0	0	0
0.0130	1	0	1	1	0	0	0	2	0	3.87	4.46	13.23	2	0	0	0	2	0
0.0061	1	0	2	1	0	0	0	2	2	4.33	4.94	12.35	2	0	0	0	2	2
0.0260	1	1	0	1	2	0	0	0	0	3.87	4.55	14.95	2	2	0	0	0	0
0.0244	1	1	1	1	2	0	0	0	2	4.31	4.99	13.63	2	2	0	0	0	2
0.0244	1	2	0	1	2	1	0	0	1	4.31	5.04	14.48	0	2	2	0	0	0
0.0013	2	0	0	1	1	0	0	0	0	3.73	4.50	17.11	2	2	0	0	0	0
0.0012	2	0	1	1	1	0	0	0	2	4.15	4.94	15.99	2	2	0	0	0	2
0.0024	2	1	0	1	1	2	0	0	0	4.15	4.99	16.83	2	0	2	0	0	0
0.0001	3	0	0	1	1	1	0	0	0	3.95	4.94	20.04	2	2	0	0	0	0
									$\mathbb{E}[\star]=$	3.64	4.11	11.41						

재고가 충분하지 않은 상태를 반영하는 것으로 보인다. 예를 들면 자산 가치가 적은 자산 1에 대한 장사정포의 공격에 대해서는 요격유도탄의 재고가 있음에도 공격 벡터의 구성에 관계없이 ADP 알고리즘은 이에 대한 방어를 전혀 하지 않는다.

2.5.3 시나리오 3: $R_t = (12,12)$

Table 3은 방어자가 2개의 포대에서 각 12회(144발)의 요격 유도탄 재고가 있는 경우에 장사정포의 공격 벡터에 따른 의사결정 결과를 보여준다.

ADP 알고리즘을 사용하는 경우 SSL을 적용한 대응 배치의 자산 손실에 비하여 11.41% 감소한 결과를 나타냈다. 이 시나리오는 적의 공격에 대하여 방어자의 요격유도탄을 충분하게 보유한 경우로 가치가 큰 자산 2에 대하여 가용한 4회의 방어를 수행한 결과를 확인할 수 있었다. 시나리오 1, 2와 비교하여 상대적으로 가치가 낮은 자산 1의 방어에도 1회 요격유도탄을 사용하고 있다.

2.5.4 분석 평가

ADP 알고리즘을 활용하여 시나리오 1, 2, 3을 시뮬레이션 한 결과는 다음과 같이 정리할 수 있다.

첫째, MDP의 경우 동적계획법을 사용하면 최적화를 위해 매우 많은 시간이 소요되는 것으로 알려져 있으나 강화학습 기반의 ADP 알고리즘을 사용할 경우 1초 이내의 매우 빠른 시간 안에 근사 최적해를 구할 수 있음을 확인하였다.

둘째, 장사정포 공격 시 요격유도탄이 부족한 경우(시나리오 1 및 시나리오 2)와 충분한 경우(시나리오 3)에 대한 방어전략의 변화를 Table 1, 2, 3을 통하여 확인하였다.

셋째, ADP를 적용하는 경우 요격유도탄의 재고에 따라 방어전략을 유연하게 적용하므로 공격 벡터에 관계없이 고정된 방어전략을 갖는 SSL보다 자산 가치의 손실이 항상 적었다.

Ⅲ. 결 론

본 논문에서는 저고도 궤적의 동시 다발성 장사정포 공격을 방어하는 문제를 강화학습 기반의 알고리즘을 활용하여 근사 최적해를 구할 수 있는 ADP를 적용하였다. 짧은 시간에 최적해 산출이 불가능한 MDP 기반의 DWTA 문제를 ADP 알고리즘을 적용하여 수 초 내에 근사 최적해를 산출하였으며, 기존에 연구된 탄도미사일 방어뿐만 아니라 저고도 궤적의 장사정포에 의한 공격 대응에도 효과적으로 사용할 수 있음을 확인하였다.

본 연구의 결과를 향후 구축 예정인 장사정포 위협에 대비한 방어 요격체계 개발에 적용한다면 효과적인 방공망 설계에 기여할 수 있을 것으로 판단된다. 아울러 방어 자산의 수 및 가치, 요격 유도탄 포대의 수 및 요격 유도탄의 재고 수준 등을 변화시켜가며 시뮬레이션을 한다면 장사정포 공격에 대한 준비태세 확립에 도움이 될 것으로 보인다.

References

1) Choi, Y. H., Lee, Y. H. and Kim, J. E., “Comparative Study on Performance of Meta-heuristics for Weapon-Target Assignment Problem,” *Journal of the Korea Institute of Military Science and Technology*, Vol. 20, No. 3, 2017, pp. 441~453.

2) Chosun media, https://www.chosun.com/politics/politics_general/2022/02/21/6SGFMAEIZZGG7PIZAMLVZ33HCU/. (accessed 2022/02/21)

3) *Defense White Paper*, Ministry of National Defense, 2014, pp. 1~332.

4) DEMA, https://kookbang.dema.mil.kr/newsWeb/m/20210720/1/BBSMSTR_000000100115/view.do?nav=0&nav2=0. (accessed 2022/02/21)

5) Park, S. G. and Lee, K. H., “A Study on the Establishment of Capability-Based Multi-Layered Missile Defense System Considering MD in U.S.,” *Journal of the KNST*, Vol. 3, No. 1, 2020, pp. 46~55.

6) Yonhapnews, https://m.yna.co.kr/view/AKR20210628082800504?input=1195m. (accessed 2022/02/21)

7) Hong, D. W., Yim, D. S. and Choi, B. W., “Application and Determination of Defended Footprint Using a Simulation Model for Ballistic Missile Trajectory,” *Journal of the Korea Institute of Military Science and Technology*, Vol. 21, No. 4, 2018, pp. 551~561.

8) Jang, J. G., Kim, K., Choi, B. W. and Suh, J. J., “A Linear Approximation Model for an Asset-based Weapon Target Assignment Problem,” *Journal Society of Korea Industrial and System Engineering*, Vol. 38, No. 3, 2015, pp. 108~116.

9) Kim, J. H., Kim, K., Choi, B. W. and Suh, J.J., “An Application of Quantum-inspired Genetic Algorithm for Weapon Target Assignment Problem,” *Journal Society of Korea Industrial and System Engineering*, Vol. 40, No. 4, 2017, pp. 260~267.

10) Lloyd, S. P. and Witsenhausen, H. S, “Weapon Allocation is NP-Complete,” *Proceedings of the IEEE Summer Computer Simulation Conference*, 1986, pp. 1054~1058.

11) Ha, M. R. and Choi, J. W., “Simulator Design for Static Weapon Allocation Algorithm Evaluation,” *Journal of Institute of Control, Robotics*

and Systems, Vol. 25, No. 4, 2019, pp. 340~345.

12) Yoon, M. H., Park, J. H., Yi, J. H. and Koo, B. J., "An Effective Weapon Assignment Algorithm in a Multi-Target and Multi-Weapon Environment," *Korea Information Science Society, Winter Conference*, 2016, pp. 87~89.

13) Choi, B. W., Yoo, B. C., Kim, J. H. and Yim, D. S., "A Study on the Flight Trajectory Prediction Method of Ballistic Missiles," 2020 *Autumn Conference on Journal of Korean Society of Systems Engineering*, 2020, pp. 131~140.

14) Yoo, B. C., Kim, J. H., Kwon, Y. S. and Choi, B. W., "A Study on the Flight Trajectory Prediction Method of Ballistic Missiles," *Journal of Korean Society of Systems Engineering*, Vol. 16, No. 2, 2020, pp. 131~140.

15) Im, J. S., Yoo, B. C., Kim, J. H. and Choi, B. W., "A Study of Multi-to-Majority Response on Threat Assessment and Weapon Assignment Algorithm: by Adjusting Ballistic Missiles and Long-Range Artillery Threat," *Journal of Korean Society of Industrial and systems Engineering*, Vol. 44 N0. 4, 2021, pp. 43~52.

16) Yook, J. K., Hwang, S. J. and Kim, T. G., "Impact of MOPs on Effectiveness for M-to-M Engagement with the Counter Long Range Artillery Intercept System", *Journal of Korean Society of Simulation*, Vol. 29, No. 3, September 2020, pp. 57~72.

17) Shin, M. K., Park, S.-S., Lee, D. and Choi, H.-L., "Mean Field Game based Reinforcement Learning for Weapon-Target Assignment", *Journal of the Korea Institute of Military Science and Technology*, Vol. 23, No. 4, 2020, pp. 337~345.

18) Davis, M. T., Robbins, M. J. and Lunday, B. J., "Approximate Dynamic Programming for Missile Defense Interceptor Fire Control," *European Journal of Operational Research*, 259, 2017, pp. 873~886.

19) Summers, D. S., Robbins, M. J. and Lunday, B. J., "An Approximate Dynamic Programming for Comparing Firing Policies in a Networked Air Defense Environment," *Computers & Operations Research 117, 2020 : 104890.*

20) Powell, W. B., *Approximate Dynamic Programming: Solving the Curse of Dimensionality*, 2011, Second Edition, John Wiley & Sons, Hoboken, NJ.

21) Powell, W. B., "Perspectives of Approximate Dynamic Programming," *Annals of Operations Research*, Vol 13, No. 2, 2012, pp. 1~38.

22) McKenna, R. S., Robbins, M. J., Lunday, B. J. and McCormack, I. M., "Approximate Dynamic Programming for the Military Inventory Routing Problem," *Annals of Operationd Research*, Vol. 288, No. 1, 2020, pp. 391~416.

23) Bradtke, S. J. and Barto, A. G, "Linear Least-Squares Algorithms for Temporal Difference Learning," *Machine Learning*, Vol. 22, No. 1, 1996, pp. 33~57.