Development of Active Decoy Guidance Policy by Utilising Multi-Agent Reinforcement Learning

Enver Bildik*, Burak Yuksek†, Antonios Tsourdos‡, Gokhan Inalhan§
Cranfield University, United Kingdom, MK43 0AL

In this paper, an AI-driven algorithm is applied to design a decoy deployment strategy which aims to increase the miss distance between the naval target and missile threat. In this scenario, three decoys are deployed from the target ship, and each decoy owns an onboard jamming system which is utilized to create an artificial scatter point. Then, an Equivalent Scattering Centre (ESC) point is formed along the target-to-decoy line which represents the radar cross-section of the group of decoys and target. The decoys are trained by utilising Multi-Agent Deep Deterministic Policy Gradient algorithm for cooperative guidance which aims to move the ESC point as far as possible from the target to increase the miss distance. Preliminary results show that the proposed approach is promising to increase the survival probability of a naval platform against an anti-ship missile that is equipped with radar seeker.

I. Introduction

Advanced missile systems equipped with the latest radar seeker technology cause a significant threat to naval platforms which have limited manoeuvring capability. To protect naval platforms in a hostile environment, defence strategies have been developed under two main approaches. While the hard-kill option aims to destroy detected threats physically, soft-kill option deceives approaching threats by manipulating them. Decoys are Electronic Countermeasure (ECM) systems widely employed as a soft-kill strategy to deceive radar seekers that direct missiles at the terminal stage. In addition to this, decoy deployment angle and deployment time are crucial parameters which determine the success of a decoying mission. A remarkable number of research has been conducted to develop effective decoying strategies. However, novel strategies are vitally important to prevail against advanced radar seekers.

The effectiveness of conventional onboard ECM applied by air platforms has decreased against the RF missile system because of advances in radar seeker technology. To guarantee the defence of a friendly aircraft, the concept of off-board ECM techniques has been developed, and thus towed-decoy has been utilized [1]. Towed decoys aim to provide a remote source of RF energy from the mother platform. In [2], performance of a towed decoy deployed by an aircraft against an anti-air missile equipped with a monopulse seeker is examined. Operational parameters including the strength of reflection, the tether length, and the direction of release, are investigated before deployment. Miss distance between the real target and missile is calculated based on abovementioned parameters under different missile incoming directions. Because the towed decoy has a connection through a tether with the aircraft, independent off-board decoys are more protectors to the host platform with the flexibility of manoeuvring. In [3], active off-board decoys are ejected from the ship to emit a source of radiation that can jam the radar signal of the seeker and direct the missile away from the ship. Launch direction and launch timing are studied because they are critical parameters that determine the success of the decoying operation. In [4], a scenario is considered in which a group of UAVs fly in a close formation, and utilize jamming resources cooperatively to prevent being tracked by surface-to-Air Missile (SAM) tracking radars. In [5], a discussion is made to evaluate the feasibility of multiple autonomous vehicles for the mission of electronic attack (EA) against integrated air defence systems (IADS). Cooperative path planning and resource allocation are highlighted problems to be addressed in this context. A collaborative decoy jamming strategy is studied in [6] by employing an ensemble of small-scale UAVs to degrade the performance of inverse synthetic aperture radar (ISAR), and for coordination, a cooperative decision-making algorithm is applied. In [7], the problem of enhancing ship survival probability against enemy torpedos by using single and multiple decoys is addressed. Moreover, same-side-deployment and zig-zag deployment strategies are applied to evaluate the success of the mission. A defence scenario is studied in [8] to deceive a missile threat by performing the optimal deployment of decoys and vertical-S manoeuvre strategy

*PhD. Student, e.bildik@cranfield.ac.uk

†Postdoctoral Research Fellow, burak.yuksek@cranfield.ac.uk, AIAA Professional Member.

‡Head of Centre for Autonomous and Cyber-physical Systems, a.tsourdos@cranfield.ac.uk.

§BAE Systems Chair, Professor of Autonomous Systems and Artificial Intelligence, inalhan@cranfield.ac.uk, AIAA Associate Fellow.

simultaneously. In addition to this, an analytical expression is developed for the miss distance calculation at intercept. The survival probability of radar against anti-radiation missiles (ARMs) is evaluated in an optimal quadrangular topology in which decoys are positioned [9]. The performance of decoys in the evader-pursuer engagement scenario is examined in [10] for diverse decoy launch angles and launch times. The range of launch angles and launch times which can guarantee that the decoy stays within the field-of-view (FOV) of radar seekers is derived in [11] based on the intersection point. A simulation method is proposed in [12] to evaluate the performance of a repeater-type active decoy in a combat scenario, and also effects of circular and linear polarization of signals on the performance are analysed. In [13], based on RF specifications of the decoy, such as the antenna patterns and amplifier gains, the jamming efficiency of an active repeater decoy is evaluated. In [14], a ducted-fan flight array system is utilized for a decoy mission to enhance the protection of the target ship against anti-ship missiles (ASMs). A sequential logic algorithm based decoy deployment strategy is developed. A reinforcement learning-based algorithm (Q-learning) is applied in [15] for decoy guidance which directs the assigned decoy to the optimal location. In [16], multiple UAVs conduct the decoy mission against anti-ship missiles, and an auction-based task assignment algorithm is applied to manage effectively duties. In [17], an engagement scenario-based simulation program is developed to assess the efficiency of the decoy. Different scenarios are conducted for various RCSs and launching directions of the decoy.

The proposed approach aims to drive an AI-based decoy deployment strategy to enhance the survivability of the target ship. In this scenario, three decoys perform a cooperative manoeuvre to deceive a sea-skimming missile. The main motivation of the multi-agent decoy deployment strategy is maximizing the miss distance between the missile and the target ship. Each decoy has an onboard jamming system which is used to create an artificial scatter point. Then, an Equivalent Scattering Centre (ESC) point is created along the target-to-decoy line which represents a joint radar cross-section (RCS) point of decoys and target. Position value and RCS value of this centre point are dependent on the location and RCS value of the target and decoys. AI-based manoeuvring policy of the active decoy system is developed by utilizing the reinforcement learning method. To improve the decoy performance and increase the survivability of the target ship, a multi-agent decoy fleet is created and it is trained by Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm because of its high efficiency for cooperative tasks. The expectation from this approach is to move the ESC point away from the target as far as possible by generating optimal cooperative guidance for the decoys, and to increase the miss distance.

Main contribution of this research is to develop an AI-based decoy deployment strategy to maximize the miss distance between naval target and missile threat. Three decoys are trained by utilising the MADDPG algorithm to execute the allocated mission cooperatively. The cooperative motion of decoys increases their joint RCS level, shifting the ESP towards decoys. Thus, decoying strategy executed by three decoys can be more effective against the missile threat. Monte Carlo Simulation (10000-run) is applied to evaluate the proposed strategy, and results show that over 80% success rate is obtained when the kill distance is equal to 100 meters, and the maximum decoy speed is 30 m/sec.

Remaining of the paper is structured as follows; Section II defines the problem we have proposed an approach to sort out it in this study, and then formulates missile-target engagement equations. Section III provides information regarding reinforcement learning algorithm applied in this work, subsequently, steps are explained to train multi agents. Section IV shares results of Monte Carlo Simulations (10000-run), and a comparison study and discussion are made between results. Concluding remarks and future works are given in Section V.

II. Problem Definition

In this study, the protection of a naval platform which is under a missile threat is addressed. The envisioned scenario, as depicted in the Figure 1, is considered in which three decoys are deployed from the deck of the main platform to deceive an approaching missile threat. A rotary-wing drone equipped with a jammer is used as a decoy which is able to mimic the radar cross-section of the target ship. In this study, the RCS of the target ship is symbolized by a scattering source point called a “scattering centre”. In an engagement environment, an ASM intercepts the virtual scattering centre, not the real target ship, and as illustrated in the Figure 1, a LOS line is created. Before decoys are activated, the scattering centre just represents the RCS level of the real target, and it is located on the target. As soon as decoys are activated with the expected RCS level, the scattering centre moves away from the target ship. Consequently, the initial LOS line moves, and the risk level of the target ship tracked by the missile reduces. The position of the scattering centre depends on the RCS level and the instantaneous positions of both the target and decoys. The formed equivalent scattering centre (ESC) is calculated using Equations (1) and (2) [17].

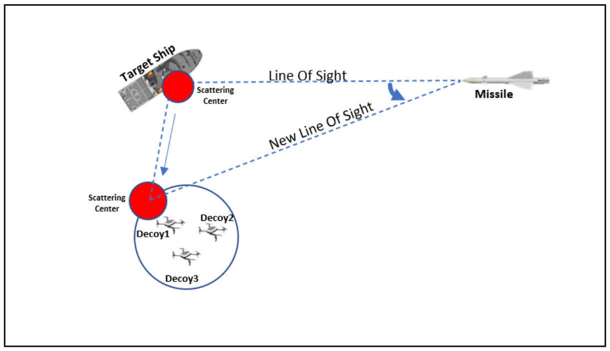


Fig. 1 The position shift of the Scattering Centre by decoys

$$\sigma_{sc} = 10\log_{10}\left(10^{\sigma_s/10} + 10^{\sigma_d/10}\right) \tag{1}$$

$$\begin{cases} x_{sc} = \frac{10^{\sigma_s/10}x_s + 10^{\sigma_d/10}x_d}{10^{\sigma_{sc}/10}} \\ \\ y_{sc} = \frac{10^{\sigma_s/10}y_s + 10^{\sigma_d/10}y_d}{10^{\sigma_{sc}/10}} \end{cases} \tag{2}$$

where σ_s, σ_d, and σ_{sc} are the RCS level in dBsm of the target ship, the joint RCS level of multi-decoy group, and the RCS level of the ESC point, respectively. (x_s, y_s), (x_{sc}, y_{sc}), and (x_d, y_d) are the instantaneous position of the ship, the ESC point, and the joint point of decoys in meter respectively. In this study, three decoys are utilized so the given formula is expanded to take into account the RCS level of each decoy.

The idea behind using multi-decoys can be explained in two points; a) the size of the payload is dependent on the signal power of the jammer, and so instead of using a single large decoy with a heavy payload, multi-small decoys with a lighter payload can be more effective, and b) in the case of any loss in multi-decoys, the rest of the decoys can execute the rest of the mission successfully.

Point mass models are used to model the kinematics of the naval platform and missile platform. The missile-target engagement is demonstrated in Figure 2. The v_T, v_M, n_T, and n_V terms represent the target velocity, missile velocity, target acceleration, and missile lateral acceleration respectively. Some fundamental equations are given below to derive the relevant parameters used for missile lateral acceleration. The R_{TM} is the distance between the target and the missile during the engagement, and it is calculated as;

$$R_{TM} = \sqrt{(R_{TMe}^2 + R_{TMn}^2)} \tag{3}$$

As seen in Figure 2, by means of trigonometry, the line-of-sight angle (λ) is found easily as;

$$\lambda = \arctan\left(\frac{R_{TMe}}{R_{TMn}}\right) \tag{4}$$

The relative velocity of the target and missile are calculated separately for each axis as below;

$$V_{TMe} = V_{Te} - V_{Me} \tag{5}$$

$$V_{TMn} = V_{Tn} - V_{Mn} \tag{6}$$

The line-of-sight rate is derived by the differentiation of line-of-sight angle equation (4). After some modifications, the value is calculated as;

$$\dot{\lambda} = \frac{R_{TMe}V_{TMn} - R_{TMn}V_{TMe}}{R_{TM}^2} \tag{7}$$

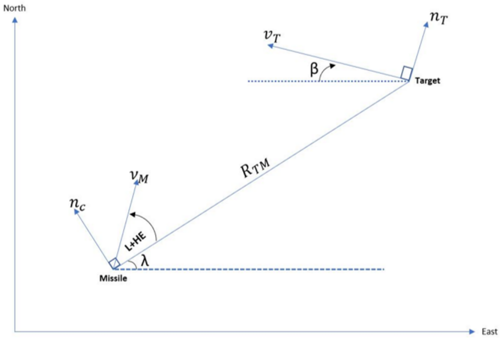


Fig. 2 Missile-Target engagement

The closing velocity is described as the negative rate of change of the missile-target range i.e R_{TM}, by applying differential equations law, it can be calculated as

$$V_c = \frac{-(R_{TMe}V_{TMe} + R_{TMn}V_{TMn})}{R_{TM}} \tag{8}$$

Lastly, by the definition of proportional navigation guidance law, the magnitude of the missile guidance command n_c is obtained as;

$$n_c = NV_c\dot{\lambda} \tag{9}$$

Here, N is constant and its value is between 3 and 5.

III. Cooperative Active Decoy Guidance Algorithm Design

A. Multi-Agent Deep Deterministic Policy Gradient

The concept behind Reinforcement Learning (RL) is based on the reward received by an agent in response to its actions during the interaction with the environment. In reinforcement learning, the environment is represented by a Markov Decision Process (MDP) with the defined state space, action space, reward function and probabilistic transition function [18]. States are the information acquired from the environment during the interaction. The action space is designated as a set of all possible actions agents can take in an environment. The agent's goal in this environment is to learn a policy which finds the best sequence of actions to maximize cumulative reward. The cumulative reward at each timestep is calculated through the Equation (3) as:

$$R(\tau) = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \gamma^3 r_{t+4} + \dots . \tag{10}$$

The γ (gamma) is a discount factor which is bounded between 0 and 1. The bigger the gamma the lesser the discount, which means the agent focuses more about the long-term reward. On the other hand, the lesser the gamma the bigger discount which means the agent considers more about the short term reward. For single-agent cases, the above-mentioned process works properly, and the agent can learn the optimal policy if a reward function is well designed. However, applying the same single-agent algorithm to train two agents together is not effective due to some reasons. At this point, Multi-Agent Reinforcement Learning (MARL) takes a role to alleviate this issue. MARL algorithms teach multiple agents to collectively learn, collaborate, and interact with each other in an environment. Non-stationary transitions and exponentially increasing state and action spaces are two main challenges MARL algorithms face. In MDPs (the structure utilized for single-agent RL), it is assumed that for each distinct state and action pair, the transition probabilities to other states remain stationary throughout. Dependent on the number of agents, sizes of state-space and action space increase exponentially, and this curse of dimensionality makes learning difficult. Plenty of approaches have been proposed to mitigate the effects of these challenges. Most of these approaches are under the umbrella of centralized training with decentralized execution frame. In [19], an actor-critic based approach, Multi-Agent Deep Deterministic Policy Gradient (MADDPG), is proposed to train multi-agents, and it develops a method to overcome the non-stationary problem.

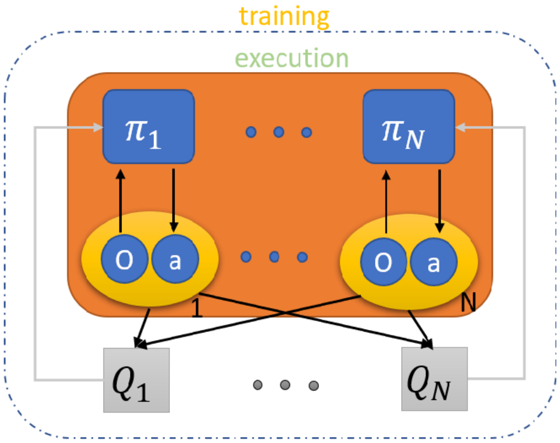


Fig. 3 Centralized training with decentralized execution framework (Adapted from [19])

In [20], a Counterfactual Multi-Agent (COMA) algorithm is proposed to address the challenges of the multi-agent credit assignment by using a counterfactual baseline. In [21], MAPPO is a modified version of the Proximal Policy Optimization (PPO) algorithm for being used for multi-agents. It is off-policy learning, and clipping is applied to stabilise the training process.

Algorithm 1: Multi-Agent Deep Deterministic Policy Gradient for N agents [19]

```
for episode = 1 to M do
    Initialize a random process $\mathcal{N}$ for action exploration
    Receive initial state x
    for t = 1 to T do
        for each agent $i$, select action $a_i = \boldsymbol{\mu}_{\theta_i}(o_i) + \mathcal{N}_t$ w.r.t the current policy and exploration
        Execute actions $a = (a_1, \ldots, a_N)$ and observe reward $r$ and new state x′
        Store $(\mathbf{x}, a, r, \mathbf{x}')$ in replay buffer $D$
        x ← x′
        for agent i = 1 to N do
            Sample a random minibatch of $S$ samples $\left(x^j, a^j, r^j, x'^j\right)$ from $D$
            Set $y^j = r_i^j + \gamma Q_i^{\mu'}(\mathbf{x}'^j, a'_1, \ldots, a'_N)\Big|_{a'_k=\mu'_k\left(o_k^j\right)}$
            Update critic by minimizing the loss $\mathcal{L}(\theta_i) = \frac{1}{S}\sum_j \left(y^j - Q_i^{\mu}\left(\mathbf{x}^j, a_1^j, \ldots, a_N^j\right)\right)^2$
            Update actor using the sampled policy gradient :
                $\nabla_{\theta_i} J \approx \frac{1}{S}\sum_j \nabla_{\theta_i}\boldsymbol{\mu}_i\left(o_i^j\right)\nabla_{a_i}Q_i^{\mu}\left(\mathbf{x}^j, a_1^j, \ldots, a_i, \ldots, a_N^j\right)\Big|_{a_i=\mu_i\left(o_i^j\right)}$
        end for
        Update target network parameters for each agent i $i$ :
            $\theta'_i \leftarrow \tau\theta_i + (1-\tau)\theta'_i$
    end for
end for
```

The MADDPG algorithm proposes an actor-critic based approach to overcome multi-agent problems inspired by its single-agent counterpart DDPG. The underlying motivation behind MADDPG is that, if the actions taken by all agents are known, the environment stays stationary even if the policies alter. DDPG is an off-policy algorithm and uses random samples from a buffer of experiences stored throughout training. Each agent owns an observation space and continuous action space. Also, every agent has four neural networks; actor, critic, target actor and target critic. The actor-network determines deterministic action based on the local observation information, while the critic network evaluates the performance of the actor by computing the Q-value. The output of the network in the actor gives directly the action result mapped with states, instead of giving a probability distribution. The framework of centralized training with decentralized execution utilized in this algorithm is depicted in Figure 3, and this framework lets the policies use extra information to facilitate training. In this proposed method, the actor only can access local information, while the critic is augmented with additional information about other agents. The parameters which are the joint state, the next joint state, the joint action, and the reward received by each agent are stored in the experience buffer for each step. A batch of random samples taken from the experience replay buffer is utilized to train agents.

B. Training of Multi Agents through MADDPG

The objective of this section is to describe steps have been followed to train 3 agents together. Before passing to training session, it is worth to know assumptions made in study. Assumptions are as follow:

Assumption 1 *Onboard sensors at the ship can determine characteristic specifications of the approaching threat such as field of view of radar seeker.*

Assumption 2 *A point mass model is utilized for the decoys, missile and naval target.*

Assumption 3 *The field-of-view angle of the missile is not constant during training. A Gaussian Noise is added to increase the uncertainty of the environment. Bounded uncertainty is defined for this information as it may not be possible to know the FOV angel of the approaching missile.*

The observation vector $\mathbf{O} \in \mathbb{R}^8$ for each agent holds information about the status of whether each decoy is within the field of view, the angle between each agent and the missile, and field of view angle. The observation vector of the *i*th decoy is given in Eq. (4)

$$\mathbf{O} = \left[D_{1_f}, D_{2_f}, D_{3_f}, MD_1, MD_2, MD_3, Up_{fov}, Low_{fov}\right] \tag{11}$$

Here, D_{i_f} depicts the current status of decoy{i} regarding whether it is within the FOV or not. D_{i_f} takes 1 if the decoy{i} is in FOV, otherwise 0. MD_i represents the line of sight angle between the missile and decoy{i}, this value is normalized between -1 and +1. Lastly, Up_{fov} and Low_{fov} indicate the angle (in radians) of upper line and lower line of the field of view area respectively. The reward function is composed of the local reward and team (formation) reward. While each agent tends to maximize its local reward, the common team reward focus on encouraging the cooperation of agents. A heuristic method is used to form a well-designed reward function, which may guarantee the convergence of the learning curve. Reward function is a linear summation of local reward and formation reward, and it is formulated as

$$r_t = r_{local} + r_{formation} \tag{12}$$

$$r_{local} = \omega_1 r_1 + \omega_2 r_2 + \omega_3 r_3 + \omega_4 r_4 \tag{13}$$

The r_{local} sub-function aims to rise the miss distance between missile and real target while guaranteeing that the decoy is within the FOV, and the reward is equal to the sum of the product of the terms with their coefficients. Here,

$$r_1 = \frac{\text{RDM}}{10000}, \quad r_2 = \frac{\text{RCM}}{10000}, \quad r_3 = \frac{\text{RCT}}{2000}, \quad r_4 = FOV, \tag{14}$$

and each $r_{\{i\}}$ term is bounded between 0 and 1 by normalization. Coefficients ω_1, ω_2, ω_3, and ω_4, are four constant values to shape the reward function, and they are given below.

$$\omega_1 = -0.5, \quad \omega_2 = -0.1, \quad \omega_3 = 0.5, \quad \omega_4 = 1 \tag{15}$$

The $r_{formation}$ purposes maintaining close flight formation for decoys and securing collision-free flight during the mission execution. To visualize the circle flight formation, below circles are drawn. Decoys should be outside of blue circle but inside of red circles, otherwise a small portion of penalty is given. The diameter of red circle is 30 meters while the diameter of the blue circle is 10 meters.

$$r_{formation} = \omega_1 r_1 + \omega_2 r_2 + \omega_3 r_3 \tag{16}$$

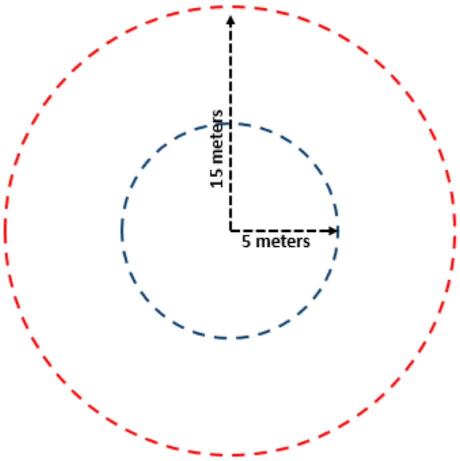


Fig. 4 Cooperative and collision-free path boundaries

$$\omega_1 = 1/3, \quad \omega_2 = 1/3, \quad \omega_3 = 1/3, \tag{17}$$

$$r_1 = \begin{cases} 0.05, & \text{if RDC} < 15 \\ -0.1, & \text{otherwise} \end{cases} \quad r_2 = \begin{cases} 0.05, & \text{if RDC} > 5 \\ -0.1, & \text{otherwise} \end{cases} \quad r_3 = \begin{cases} 0.05, & \text{if } 10 <= distance <= 30 \\ -0.1, & \text{otherwise} \end{cases} \tag{18}$$

The RDC represents the distance between each decoy and the central point of the circular formation topology. The r_1 and r_2 check whether the decoys are inside or outside of the pre-defined circles. The *distance* term indicates the range between two decoys D_iD_j, and r_3 reward aims to maintain a space between decoys for collision-free path planning. The notations RDM, RCM and RCT denote the range between the decoy and missile, the range between the scatter center and missile, and the range between the scatter center and target respectively.

The action vector $\mathbf{u} \in \mathbb{R}^2$ is given in Eq. (12), and it consists of two parameters. X_{force}, and Y_{force} represent the force (Newton) applied on the x-axis and the force applied on the y-axis respectively. The action space command is bounded as [-10, +10]. The action values are used to calculate the instant acceleration of decoys.

$$u = [X_{force}, Y_f orce] \tag{19}$$

Table 1 Initialization values of Missile, Ship and Decoys.

Parameters	Values	Units
Missile Position	(10000 ±500, 10000 ±500)	*meter*
Ship Position	(5000 ±300, 5000 ±300)	*meter*
Decoys Position	(ShipPos ±15, ShipPos ±15)	*meter*
Target heading angle	(0, 360)	*degree*

The Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm has an actor and critic neural network structure. In this study, the structures of both neural networks are identical which means they have same layer and neuron numbers. The NN structure has 2 hidden layers, each with 128 neurons. Detailed information regarding hyper-parameters is shared in the table. A rectified linear units (Relu) function is applied as an activation function for each neuron in layers, except for that in the actor output layer, tanh function is employed. While the critic network gives

Table 2 Hyper-parameters settings

Parameters	Value
Number of Agents	3
Maximum episode number	20000
Maximum step number	280
Critic learning rate	1e-3
Actor learning rate	1e-4
Experience Buffer Length	1e6
Discount Factor	0.99
Mini Batch Size	128
Sample Time	0.1

the Q-value of state-action pair, the actor network gives direct outcome as an action value.

$$g(z) = \begin{cases} z, & \text{if } z >= 0. \\ 0, & \text{if } z < 0. \end{cases} \tag{20}$$

$$g(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{21}$$

Training is done for 20000 episodes, with each episode lasting at most 280 time steps. The training of one episode stops when the range between the missile and the target is lower than the pre-defined fuse-range value. At the each reset time, some parameters are randomly initialized with respect to the bounded value ranges as given in the Table1. While the missile is randomly launched from any corner of the environment, target ship starts moving at the center of the environment. Decoys are located at the deck of the ship.

IV. Simulation Results and Discussion

The visualization of the simulation environment was created as given in the Figure 5. Area of the environment is $10 \times 10km$, and it has two different coloured rectangle regions that are predefined as initialization regions for the missile and the target. The missile system is randomly launched from one of the green zones located at each corner. Randomly assigned starting point of the target ship, which has three decoys on its deck, is initialized in the red region. In the envisaged scenario, as soon as sensors on-board the ship detect an incoming missile threat, a command is given to decoys to activate them. After that, decoys move away from the target ship cooperatively to maximize the miss distance between the missile threat and the ship platform.

At the first stage of the missile-target engagement, the radar seeker locks on a virtual Scatter Centre (SC) point, and its RCS level is only determined by the target ship. After the deployment of the multiple decoys, the number of targets (decoys and naval target) increases in the FOV of the missile radar seeker. In this instance, the ESC point is formed, and its RCS level is determined by all targets in the FOV. As the missile threat gets closer to the targets, the FOV of radar seeker narrows, so it must choose one of them to hit. If the RCS level created by multi-decoys is more effective than RCS level of the real target, the deceiving mission to the missile will be succeeded. A joint decoying RCS is generated by the close formation of decoys, and it is calculated based on the instant RCS level of each decoy. In the case of leaving the field of view area, the RCS level will be zero, leading decreasing the joint RCS level of decoys. The objective is to teach agents taking optimal actions which create a cooperative path planning, as well as guaranteeing them staying in the field of view of radar seeker. In this way, the joint RCS level of multi-decoys increases.

One important constraint that should be mentioned is the boundary of the field of view. During training and execution of the model, the field of view is limited from 4 to 6 degrees, and at each initiation, a random float number selected

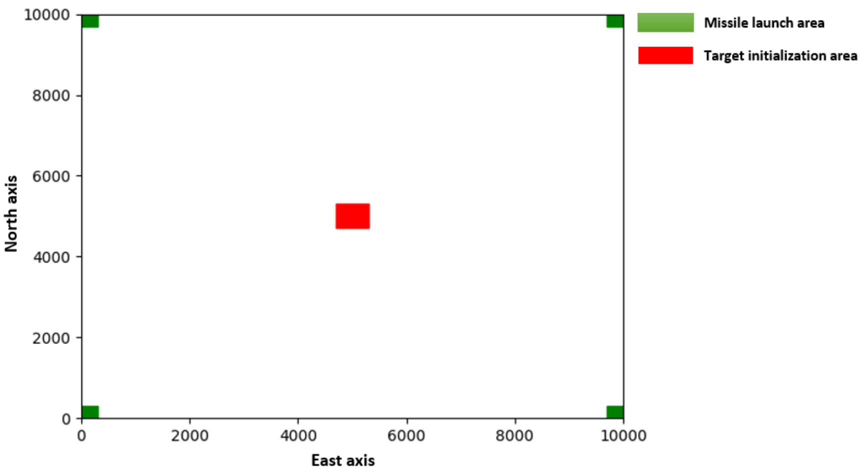


Fig. 5 Envisioned Simulation Environment

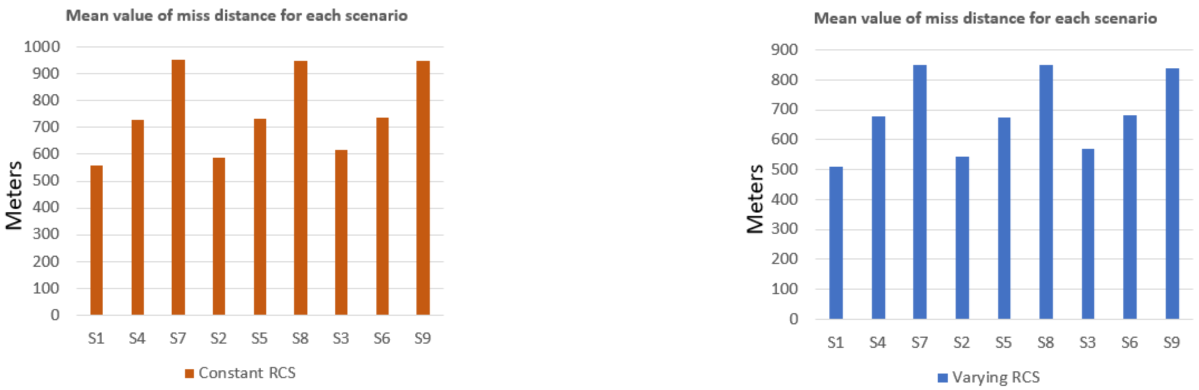
Table 3 Case1: Monte Carlo Simulation Results for Constant RCS

Kill Distance **Maximum Decoy Speed**	**100 meters**	**150 meters**	**200 meters**
20 m/sec	S1 - 82%	S2 - 74%	S3 - 64%
30 m/sec	S4 - 90%	S5 - 86%	S6 - 81%
40 m/sec	S7 - 91%	S8 - 84%	S9 - 81%

between 4 and 6 is assigned to FOV. This uncertain situation allows agents to enhance the robustness of learning. In order to examine the performance of the proposed decoy deployment strategy, a parametric study has been done. In the first case, the RCS level of the target and decoys are constant during the mission, while in the second case the RCS level of the target and decoys vary according to the defined bounded interval. Two parameters are considered for a comparison investigation: 1) kill distance, and 2) maximum decoy speed. For each case, 9 distinct scenarios are tested based on these parameters through the same generalized trained agent. A Monte Carlo Simulation (10000-run) is executed for each scenario to analyse the fundamental impact of each parameter. The mission success is calculated by comparing the miss distance with the kill distance, i.e., if the miss distance is greater than kill distance, the mission is successful.

Table 3 represents the mission success rate results (in percentage) obtained from a Monte Carlo batch simulation for the case where the RCS levels of the target and decoys are constant. S{i} denotes the sequence number of scenarios. As seen in the comparison table, the mission success rate decreases with respect to the increase in the kill distance while decoy speeds are the same. From Scenario 1 to Scenario 3, the mission success rate declines by 18%. The reason behind this decrease is that for Scenario 1, in cases where the success rate increase, the miss distance may be between 100 and 200 meters to a certain extent proportion. As a result of this, as the kill distance increase, the mission success rate decreases. To see the impact of maximum decoy speed on the mission success rate, scenarios at the same columns in the table (e.g. from Scenario 1 to Scenario 7) compare with each other because their kill distance value is the same. However, in general, an increase in the success rate can be seen with respect to increasing decoy speed except in some occasions. From Scenario 2 to Scenario 5, the mission success rate increased by 12 % but from Scenario 5 to Scenario 8, a 2 % decrease can be seen in the mission success rate. The reason behind this relation can be that because of having high speed (40 m/sec in Scenario 8), decoys move outside of the FOV of the radar seeker.

In contrast to the first case, in the second case, RCS levels of the target and decoys alter during the training and execution of the model. The RCS value of each decoy is assigned a random number between 41 and 46 (dBsm), while the RCS level of the target is allocated a random number between 45 and 50 (dBsm). The reason behind this is to evaluate the effect of varying RCS levels on the mission success rate. Table 4 depicts the mission success rate (in percentage) of the proposed decoying strategy by applying Monte Carlo simulation for the case varying RCS levels. To investigate the individual impact of the kill distance, scenarios are compared in which all have the same maximum decoy



(a) Average miss distance - Case1

(b) Average miss distance - Case2

Fig. 6 Miss distance mean value for MCS (10000 trials)

Table 4 Case2: Monte Carlo Simulation Results for Varying RCS

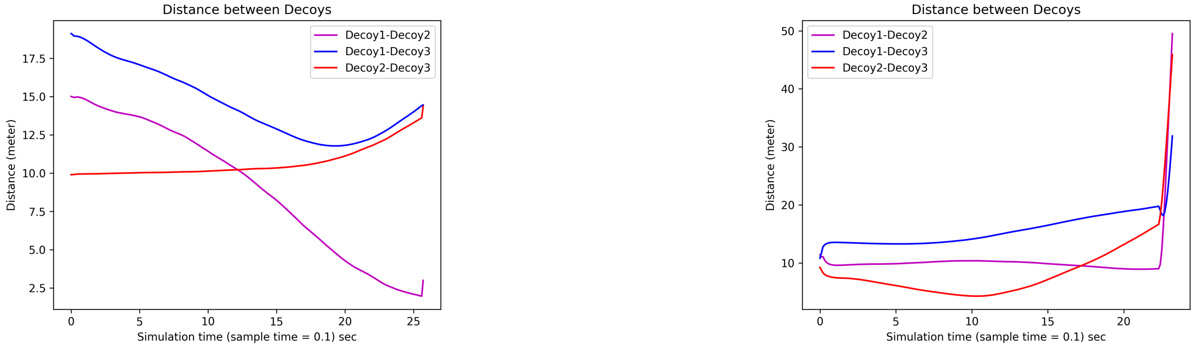
Kill Distance **Maximum Decoy Speed**	**100 meters**	**150 meters**	**200 meters**
20 m/sec	S1 - 77%	S2 - 69%	S3 - 60%
30 m/sec	S4 - 80%	S5 - 76%	S6 - 69%
40 m/sec	S7 - 79%	S8 - 74%	S9 - 70%

speed. In scenarios in the same rows, the mission success rate decreases as the kill distance value increase from 100 meters towards 200 meters. To examine the individual impact of the maximum decoy speed, scenarios are compared in which all have the same kill distance values. In scenarios in the same columns, the mission success rate increases, but not always, as the maximum decoy speed value increase from 20 m/sec towards 40 m/sec. Inferences taken from this case are almost the same as the previous case in terms of the maximum decoy speed and the kill distance. However, as can be seen in both tables, approximately 10 % differs from case 1 to case 2. The first case provides more protection to the target ship against the missile threat compared with that the second case.

To examine the impact of the maximum decoy speed on the miss distance value, scenarios which have same kill distance threshold value are compared. Figure 6 represents the mean of miss distance values obtained from 10k runs for each scenario. The maximum decoy speeds in the S1, S4, and S7 are 20, 30, and 40 m/sec, respectively. As seen in the figures, for all scenarios in two cases, the mean of miss distance values increases as the maximum decoy speed rises. The values in the Figure 6a are minor higher than that of the Figure 6b, because the mission success rate in the first cases are higher than that of the second case.

To make a fair comparison between case 1 and case 2, results of the scenario 4 in both cases, which has the same maximum decoy speed and kill distance are evaluated. The space in the deck where decoys are placed is a rectangle with one edge is 30 meters. A close formation for decoys is encouraged to increase the joint RCS level of decoys. Hence, the risk of collision of decoys arose, and to manage the close formation and collision-free path planning, a proper reward is credited. Figure 7 (a and b) depicts the relative distance between decoys during the target-missile engagement. As seen in the Figure 7b, at the last duration of the simulation, decoys move away little from each other, but this situation does not affect the mission success rate. Because, generally, after 15th seconds only decoys are within the field of view of the seeker in case the mission is successful. Namely, although decoys are far away from each other, it seems impossible for the missile to relock the target ship. In addition to this, we can compare the obtained results when maximum decoy speeds are 30 and 40 m/sec cases. there is no efficient increase in the mission success rate between 30 m/sec and 40 m/sec. However, battery duration decrease when decoy speed increase, from this, we can infer that a decoy with 30 m/sec speed can achieve a significant role in the deceiving mission.

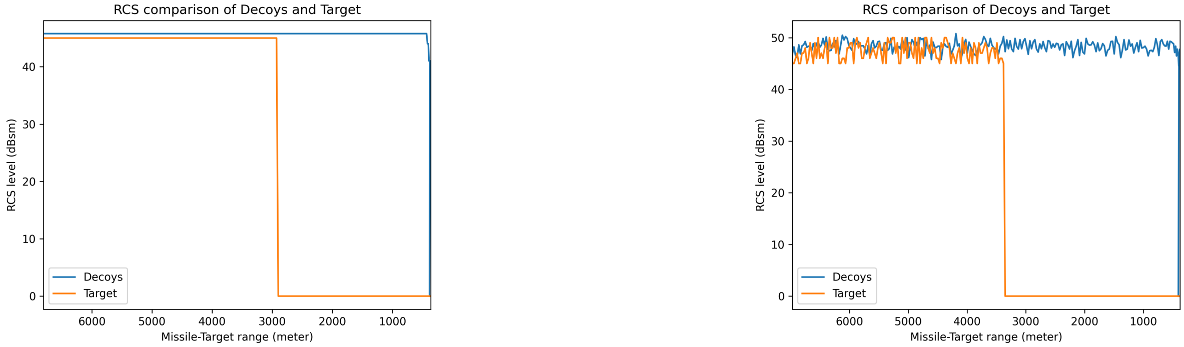
The Figure 9 depicts the miss distance distribution of the Monte Carlo Simulation (10000 trials) for case1 and case2. Green dots represent miss distance value of a run in which the mission is executed successfully, while the red dots



(a) Relative distances between decoys - Case1

(b) Relative distances between decoys - Case2

Fig. 7 Relative distances between decoys



(a) The RCS levels of the target and decoys - Case1

(b) The RCS levels of the target and decoys - Case2

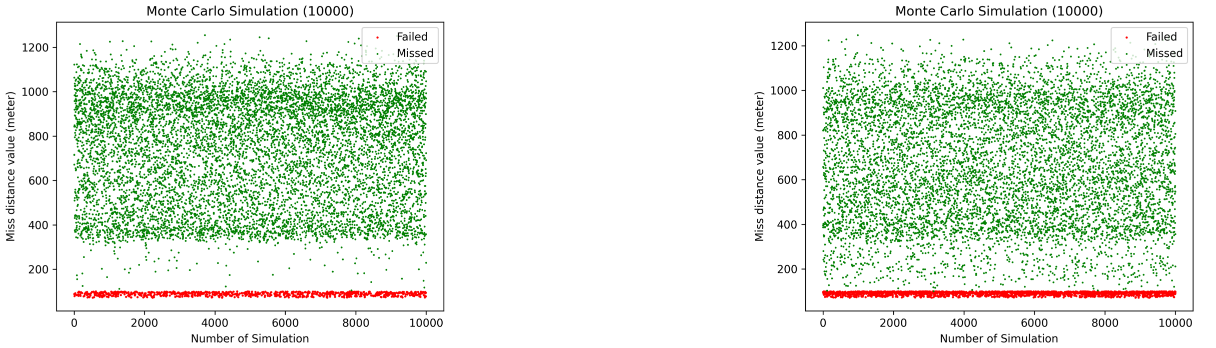
Fig. 8 The RCS levels of the target and decoys

show the miss distance value of a failed mission. The threshold of kill distance is 100 meters, which indicates the miss distance value above the threshold represents success but the versa represents failure. In both figures 9a and 9b, the density of miss distance distribution can be seen between 400 meters and 1000 meters.

The Figure 10 and 11 demonstrate the trajectories of the missile, target, and decoys when decoys execute the mission to protect the target ship. In both cases, decoys provide a high survival probability to the target ship by luring the approaching missile threat. The proposed approach can guarantee the protection of the ship in major proportion regardless of the missile threat's direction and the target's heading angle.

V. Conclusion

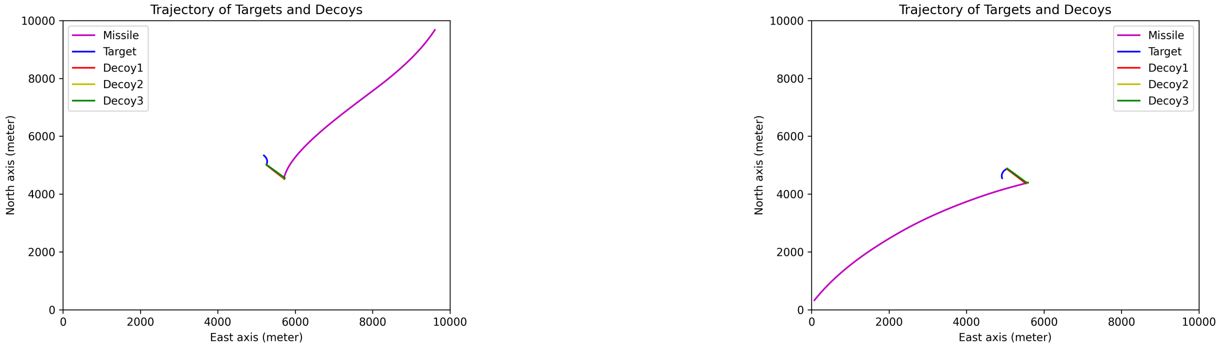
In this paper, an AI-driven decoy deployment strategy is proposed to ensure the protection of the target platform against oncoming missile threat. Point-mass model is employed for the define kinematics of the missile, target, and decoys to create two-dimensional motion. Three decoys are deployed from the main platform to eliminate missile threat approaching to the naval target. To execute this mission, decoys must be guided cooperatively. The MADDPG algorithm is applied to train decoys taking an optimal action based on the current observation of the environment. The trained generalized system is tested based on the three parameters; 1) RCS level of the target and decoys, 2) maximum decoy speed, and 3) kill distance. A Monte Carlo Simulation (10000 trials) is applied for 18 different scenarios, and the mission success rates are compared. Results shows that the proposed AI based decoying strategy can guarantee the protection of the target platform over 75 % for all scenarios when kill distance is 100 meters. In this study, the RCS level is based on a single scatter, but multi-scatter points are in consideration to calculate a joint RCS level for the target. In this case, more realistic analyses could be done for the performance of the proposed decoy deployment strategy.



(a) Miss distance value distribution - Case1

(b) Miss distance value distribution - Case2

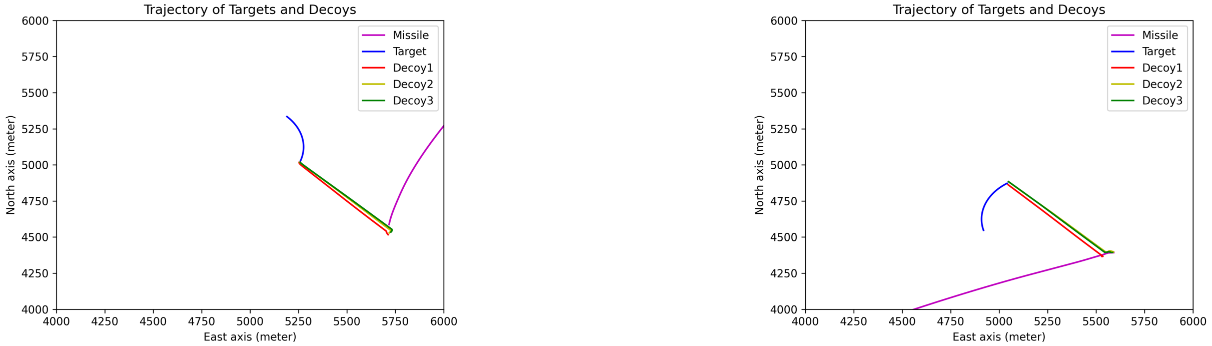
Fig. 9 Miss distance distribution of Monte Carlo Simulation



(a) trajectories - Case1

(b) Trajectories - Case2

Fig. 10 Trajectories of the missile, target, and decoys.



(a) Trajectories - Case1

(b) Trajectories - Case2

Fig. 11 Zoom views of the trajectories in Figure 10

Acknowledgments

This work is supported, in parts, by the Engineering and Physical Sciences Research Council [Grant number: EP/V026763/1].

References

[1] Kerins, W. J., "Analysis of towed decoys," *IEEE transactions on aerospace and electronic systems*, Vol. 29, No. 4, 1993, pp. 1222–1227.

[2] Yeh, J.-H., "Effects of towed-decoys against an anti-air missile with a monopulse seeker," Ph.D. thesis, Monterey, California. Naval Postgraduate School, 1995.

[3] Tan, T.-H., "Effectiveness of Off-Board Active Decoys Against Anti-Shipping Missiles." Tech. rep., NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 1996.

[4] Kim, J., and Hespanha, J. P., "Cooperative radar jamming for groups of unmanned air vehicles," *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, Vol. 1, IEEE, 2004, pp. 632–637.

[5] Mears, M. J., "Cooperative electronic attack using unmanned air vehicles," *Proceedings of the 2005, American Control Conference, 2005.*, IEEE, 2005, pp. 3339–3347.

[6] Ilaya, O., Bil, C., and Evans, M., "Distributed and Cooperative Decision Making for Multi-UAV Systems with Applications to Collaborative Electronic Warfare," *7th AIAA ATIO Conf, 2nd CEIAT Int'l Conf on Innov and Integr in Aero Sciences, 17th LTA Systems Tech Conf; followed by 2nd TEOS Forum*, 2007, p. 7885.

[7] Akhil, K., Ghose, D., and Rao, S. K., "Optimizing deployment of multiple decoys to enhance ship survivability," *2008 American Control Conference*, IEEE, 2008, pp. 1812–1817.

[8] Vermeulen, A., and Maes, G., "Missile avoidance maneuvres with simultaneous decoy deployment," *AIAA Guidance, Navigation, and Control Conference*, 2009, p. 6277.

[9] Zhou, W., Luo, J., Jia, Y., and Wang, H., "Performance evaluation of radar and decoy system counteracting antiradiation missile," *IEEE transactions on aerospace and electronic systems*, Vol. 47, No. 3, 2011, pp. 2026–2036.

[10] Ragesh, R., Ratnoo, A., and Ghose, D., "Analysis of evader survivability enhancement by decoy deployment," *2014 American Control Conference*, IEEE, 2014, pp. 4735–4740.

[11] Ragesh, R., Ratnoo, A., and Ghose, D., "Decoy Launch Envelopes for Survivability in an Interceptor–Target Engagement," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 3, 2016, pp. 667–676.

[12] Rim, J.-W., Koh, I.-S., and Choi, S.-H., "Jamming performance analysis for repeater-type active decoy against ground tracking radar considering dynamics of platform and decoy," *2017 18th International Radar Symposium (IRS)*, IEEE, 2017, pp. 1–9.

[13] Rim, J.-W., and Koh, I.-S., "Effect of beam pattern and amplifier gain of repeater-type active decoy on jamming to active RF seeker system based on proportional navigation law," *2018 19th International Radar Symposium (IRS)*, IEEE, 2018, pp. 1–9.

[14] Jeong, J., Yu, B., Kim, T., Kim, S., Suk, J., and Oh, H., "Maritime application of ducted-fan flight array system: Decoy for anti-ship missile," *2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*, IEEE, 2017, pp. 72–77.

[15] Rajagopalan, A., "Active Protection System Soft-Kill Using Q-Learning," *International Conference on Science and Innovation for Land Power, Australia Defence Science and Technology*, 2018.

[16] Dileep, M., Yu, B., Kim, S., and Oh, H., "Task Assignment for Deploying Unmanned Aircraft as Decoys," *International Journal of Control, Automation and Systems*, Vol. 18, No. 12, 2020, pp. 3204–3217.

[17] Kim, K., "Engagement-Scenario-Based Decoy-Effect Simulation Against an Anti-ship Missile Considering Radar Cross Section and Evasive Maneuvers of Naval Ships," *Journal of Ocean Engineering and Technology*, Vol. 35, No. 3, 2021, pp. 238–246.

[18] Sewak, M., *Deep reinforcement learning*, Springer, 2019.

[19] Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Pieter Abbeel, O., and Mordatch, I., "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, Vol. 30, 2017.

[20] Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S., "Counterfactual Multi-Agent Policy Gradients," 2017.

[21] Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., and Wu, Y., "The Surprising Effectiveness of PPO in Cooperative, Multi-Agent Games," *arXiv preprint arXiv:2103.01955*, 2021.