IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 January 2024, accepted 17 January 2024, date of publication 24 January 2024, date of current version 5 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3358211

RESEARCH ARTICLE

Reinforcement Learning-Based Counter Fixed-Wing Drone System Using GNSS Deception

MYOUNG-HO CHAE[1,2], SEONG-OOK PARK[1], (Senior Member, IEEE), SEUNG-HO CHOI[2], AND CHAE-TAEK CHOI[2]

[1]Korea Advanced Institute of Science and Technology, Daejeon 34141, Republic of Korea
[2]Agency for Defense Development, Daejeon 34316, Republic of Korea

Corresponding author: Seong-Ook Park (soparky@kaist.ac.kr)

This work was supported by the Agency for Defense Development (ADD) under Grant 912759101.

ABSTRACT As drone intrusions into important facilities have increased, research on drone countermeasures has been conducted to counter drones. In this study, we developed a reinforcement learning (RL)-based counter fixed-wing drone system that can respond to fixed-wing drones in autonomous flight with soft kills. The system redirects fixed-wing drones to a designated target position using the global navigation satellite system (GNSS) deception based on the drone's position and speed measured by RADAR. In this study, to construct an environment for training an RL agent, simplified drone modeling was performed for two types of fixed wing drones, and the RADAR error measured through flight tests was modeled. Subsequently, the Markov decision process (MDP) was defined to enable redirection without prior information regarding fixed-wing drones. After applying the RL agent trained in the defined MDP and environment to the counter fixed-wing drone system, the simulation and flight test results confirmed that redirection was possible for both types of fixed-wing drones.

INDEX TERMS Anti-drone system, electronic countermeasures, GNSS deception, reinforcement learning.

I. INTRODUCTION

Cases of illegal drone intrusion also increase along with the increase in the use of drones [1], [2]. Therefore, anti-drone research has been conducted to protect critical facilities from illegal drones [3], [4], [5], [6]. Anti-drone technology detects drones and responds to them to prevent them from carrying out their missions. It is divided into drone detection and drone countermeasures [7], [8]. Drone detection involves the detection and tracking of drones using RADAR, radio frequency (RF) scanners, Electro-Optical/Infra-Red (EO-IR), and acoustic sensors. Drone countermeasures are divided into two types: (1) hard kill, which disrupts the drone's mission by physically damaging it with lasers, netting, or shooting, and (2) soft kill, which uses jamming to disrupt the mission. Between the two, the soft-kill method interferes with the drone's remote-control signal or global navigation satellite system (GNSS) signal. The autonomous flight of commercial drones, which have limited high-performance inertial measurement units (IMU), relies on GNSS [9], [10], [11]. Therefore, research has been conducted on soft kill for GNSS, interfering with the automatic flight of drones by interfering with or deceiving the reception of GNSS signals [12], [13], [14].

Particularly, research has been conducted on using GNSS deception to cause autonomous drones to deviate from their path or to redirect near a designated target position. These were studied separately based on the type of drone, flight mode, and use of path-following algorithms [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25]. Multi-rotor drones have the advantages of easy takeoff and landing, hovering, and excellent maneuverability, whereas fixed-wing drones are characterized by high flight speeds and long flight times, resulting in a wide operating distance. Therefore, multi-rotor

The associate editor coordinating the review of this manuscript and approving it for publication was Xuebo Zhang.

and fixed-wing drones are used differently depending on the mission, and a response to both drones is necessary for the drone defense of important facilities. The redirection of multi-rotor drones has been validated through flight tests, mainly for the hovering mode [15], [18], [19], [20], [21]. In [25], the redirection of fixed-wing drones was verified through simulations and flight tests.

In contrast, reinforcement learning (RL)-based redirection using a hunter drone capable of meaconing has been proposed to counter multi-rotor drones [26]. Meaconing is a soft kill that retransmits the received GNSS signal. For synchronized deception, the hunter drone must be within one chip of the intruder drone. Moreover, for practical implementation, it is necessary to eliminate the ring around. GNSS deception using RADAR performs synchronized deception based on the position of the drone measured by the RADAR. Compared to meaconing, this enables it to operate on the ground.

Reinforcement learning requires time to build an environment for RL agent and for training, but it is known to perform better than existing rule-based methods for a given dynamic environment [27]. Previous research on redirection using GNSS deception has used rule-based methods; however, to the best of our knowledge, no research has been conducted using RL. In this study, we designed and verified an RL-based counter fixed-wing drone system that can perform fixed-wing drone redirection using GNSS deception. Modeling of the environment is required for RL agent training; for this purpose, drone and RADAR modeling were performed. It was confirmed that the simplified fixed-wing drone model has flight trajectories and tendencies similar to the flight test results in [25] and that this drone model can quickly model a drone with fewer tuning parameters than a detailed modeled drone. In addition, it performs simulations faster than a real-time drone simulator. This enables the Markov decision process (MDP) for RL agent training to be defined and the performance of the trained RL agent to be verified more quickly. Therefore, in this study, simplified drone modeling was performed for two types of fixed-wing drones. However, the verification of the actual operating environment is necessary because of the difference between the trained RL agent environment and the actual operating environment. Therefore, the trained RL agent was applied to a counter fixed-wing drone system and its performance was verified through flight tests on two types of fixed-wing drones. The contributions of this study are as follows:

(i) A method for training the RL agent of an RL-based counter fixed-wing drone system is proposed using simplified fixed-wing drone models.
(ii) The effectiveness of the proposed training method is confirmed by verifying the redirection of the trained RL agent through flight test results using two types of fixed-wing drones.

The remainder of this paper is organized as follows: Section II presents a counter fixed-wing drone system using RADAR and GNSS spoofer. Section III describes RL for fixed-wing drone redirection, including the environmental configuration and MDP definitions. Section IV presents the simulation results of the RL-based counter fixed-wing drone system using two simplified drone models. Section V presents the redirection results of the system based on flight tests conducted using two types of drones. Section VI presents a discussion of the results compared with a rule-based approach. Finally, Section VII concludes the paper.

II. COUNTER FIXED-WING DRONE SYSTEM USING RADAR AND GNSS SPOOFER

A. COUNTER FIXED-WING DRONE SYSTEM USING GNSS DECEPTION

Fig. 1 shows the counter fixed-wing drone system using GNSS deception, which consists of a GNSS spoofer and RADAR in a closed-loop structure. The RADAR measures the position and speed of the drone and feeds them into the Kalman filter of the GNSS spoofer, converting the aperiodic input of the RADAR into a periodic output and performing filtering. This is input into the redirection algorithm, which calculates the deception position and velocity to redirect the drone to the designated target position, after which a deception signal is generated and radiated to the drone. If the flight direction of the drone is changed by the deception signal, the RADAR measures it again and repeats the operation to redirect the drone to a designated target position.

The counter fixed-wing drone system can measure the position of the target drone using RADAR, which is close to the position of the GNSS receiver mounted on the drone. Therefore, soft spoofing, which is synchronized GNSS deception, can be performed using the procedure shown in Fig. 2. First, the GNSS spoofer receives an authentic signal from the satellites and the measured position of the target GNSS receiver from the RADAR. Then, the spoofer performs time-delay control on the authentic signal of each received satellite and radiates the spoof signal with a power greater than the authentic signal, such that the target GNSS receiver recognizes the position measured by the RADAR. As shown in the middle of Fig. 2, the target GNSS receiver receives the deception signal reflecting the RADAR measurement error and the spoofer's time delay error, and tracks the deception signal because it is adjacent to the authentic signal and has a higher power than the authentic signal.

Subsequently, the time delay value for each satellite for the desired deception position was calculated and controlled to perform synchronized dynamic deception.

Synchronized dynamic deception can be used to generate the deception position and velocity for drone redirection. A real-time controllable GNSS spoofer is required to redirect the drone in real time.

B. REAL-TIME CONTROLLABLE GNSS SPOOFER

The GNSS spoofer was controlled in real time using the operational procedure shown in Fig. 3. After receiving the authentic signal $S_{L1}(t)$ from the satellites in (1), navigation message extraction and time synchronization are performed.

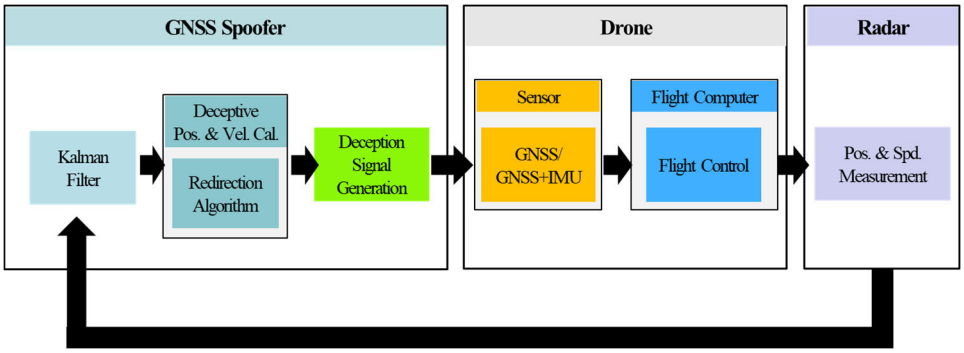

FIGURE 1. Configuration of counter fixed-wing drone system using GNSS deception.

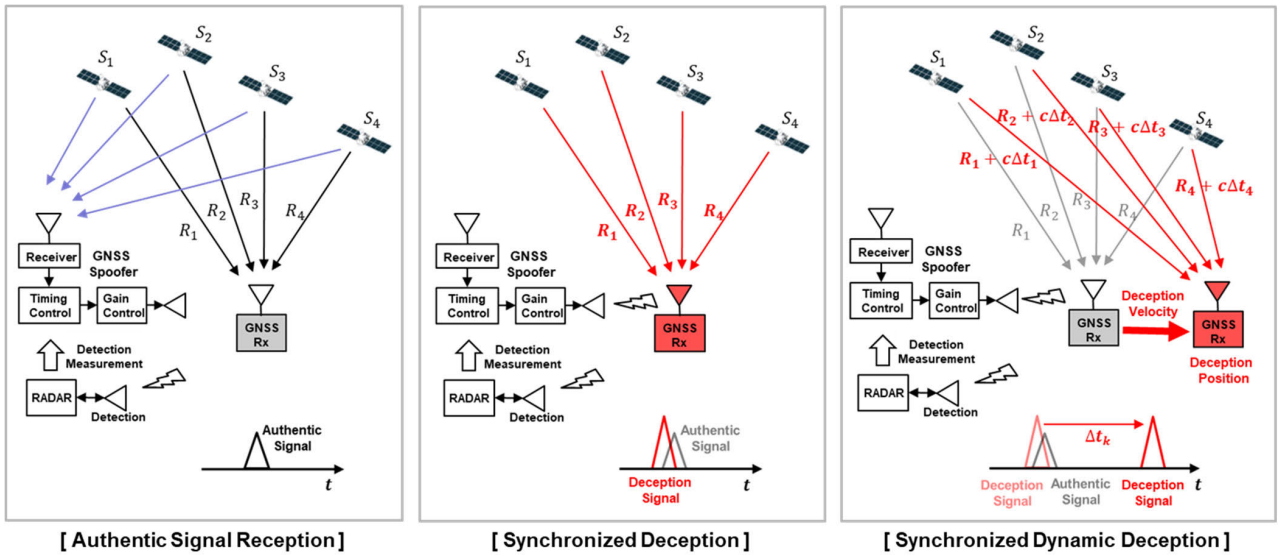

FIGURE 2. Synchronized GNSS deception for soft spoofing.

Then, the time-delay value Δt_{d_i} of each satellite signal is calculated, as shown in (2), based on the deception position X_{SP} calculated by the redirection algorithm and the measured drone position X_M from RADAR.

$$S_{L1}(t) = \sum_{j=1}^{n} A_j(t)\, G_j(t)\, D_j(t) \sin\left(\omega t + \phi_j\right), \quad (1)$$

$$\Delta t_{d_j} = \frac{\left\|X_{SP} - X_{Sat_j}\right\|_2 - \left\|X_M - X_{Sat_j}\right\|_2}{c} \quad (2)$$

where $S_{L1}(t)$ represents the sum of the signals from the n satellites, consisting of signal strength $A_j(t)$, PRN code $G_j(t)$, navigation message signal $D_j(t)$, angular frequency ω, and phase ϕ_j, which are time-dependent variables. The subscript j represents the j-th satellite. The position X_{Sat_j} of the j-th satellite signal is calculated from the ephemeris information in the navigation message, and the speed of light

Then, after calculating the time delay $\Delta t_{d_j}^{SP}$ at the GNSS spoofer, as shown in (3), the deception signal $S'_{L1}(t)$ is generated, as shown in (4). The time-delay calculation and deception signal generation are repeated for each time step.

$$\Delta t_{d_j}^{SP} = \Delta t_{d_j} - \Delta t_p - \Delta t_{sys} - \Delta t_{ion_j} - \Delta t_{tro_j} \quad (3)$$

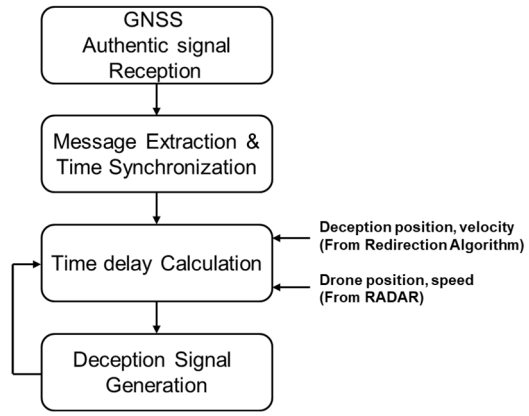

FIGURE 3. Operation process of real-time controllable GNSS spoofer. c is used to calculate Δt_{d_j}.

$$S'_{L1}(t) = \sum_{j=1}^{n} A'_j(t'_j) G_j\left(t'_j\right) D_j\left(t'_j\right) \sin\left(\left(\omega + \Delta\omega_j\right) t'_j\right. \quad (4)$$

Considering the transmission delay Δt_p, the system delay Δt_{sys} of the counter fixed-wing drone system, the ionosphere delay Δt_{ion_j}, and the troposphere delay Δt_{tro_j}, the

time delay $\Delta t_{d_j}^{SP}$ at the GNSS spoofer can be calculated. The real-time controllable GNSS spoofer was fabricated using a GNSS receiving antenna, navigation message extractor, time synchronization receiver, RF front-end, FPGA, DSP, and GNSS transmitting antenna. DSP and FPGA are synchronized through the 1 pulse per second and reference clock generated by the synchronization receiver. DSP calculates $\Delta t_{d_j}^{SP}$ and $\Delta\omega_j$ using deception information from the redirection algorithm and the extracted navigation message, and FPGA generates a deception signal by applying $\Delta t_{d_j}^{SP}$ and $\Delta\omega_j$ to each satellite. After combining the deception signals of each satellite, the signal is normalized to prevent saturation of the DAC. The frequency of the deception signal is then upconverted at the RF front end, and the signal is radiated through the antenna. Furthermore, the spoofer stops receiving authentic signals and uses the stored navigation messages when transmitting deception signals to avoid self-spoofing. Signal reception is performed again when the signal transmission ends. Therefore, a GNSS spoofer was designed to generate the deception position and velocity, which were calculated using the redirection algorithm in real time.

C. SIMPLIFIED DRONE MODEL

In this study, a simplified drone model was used to design an RL-based counter fixed-wing system. The simplified drone model has the advantage of simple drone modeling with various flight characteristics and few tuning parameters. Because the redirection was only performed along the horizontal axis, the drone model was also modeled along the horizontal axis, and the attitude controller was omitted.

Carrot chasing, one of the path-following algorithms, calculates the cross-track error based on the drone position and sets virtual target point (VTP) to the point along the path line added by δ at the intersection of the path line and cross-track error, as depicted in Fig. 4. $\vec{W}_{i+1}$ and $\vec{W}_i$ represent the vectors of the i+1-th and i-th waypoints, respectively. $\vec{VTP}_k$ is the position vector of VTP, and $[x_k^{VTP}, y_k^{VTP}]^T$ are its position coordinates. $[x_k^{sensor}, y_k^{sensor}]^T$ are the coordinates of the position measured by the drone sensor, and $\vec{D}_k^{sensor}$ is the position vector measured by the sensor. $\vec{R}_i$ represents the path line vector connecting $\vec{W}_{i+1}$ and $\vec{W}_i$, and $\vec{D}_k$ represents the position vector of the drone. The variable δrefers to the distance at which VTP is set. $\vec{V}_k^{desired}$ represents the desired velocity vector, $\dot{\psi}_k^{desired}$ indicates the desired heading rate, and $\vec{V}_k^{sensor}$ represents the velocity measured by the drone sensor. Subscript k of the variable indicates the k-th time step.

The drone model calculates the desired heading $\psi_k^{desired}$ to fly toward the VTP for path-following, as in (5) and (6).

$$\vec{V}_k^{desired} = \left(\begin{bmatrix} x_k^{VTP} \\ y_k^{VTP} \end{bmatrix} - \vec{D}_k^{sensor} \right) T^{-1} \tag{5}$$

$$\psi_k^{desired} = atan2\left(\vec{V}_{k,y}^{desired}, \vec{V}_{k,x}^{desired} \right) \tag{6}$$

where T indicates the sample time.

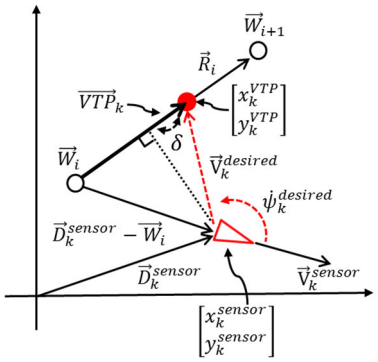

FIGURE 4. Simplified fixed-wing drone model.

The current measured heading ψ_k^{sensor} and $\dot{\psi}_k^{desired}$ of the drone are calculated using (7) and (8), respectively.

$$\psi_k^{sensor} = atan2\left(\vec{V}_{k,y}^{sensor}, \vec{V}_{k,x}^{sensor} \right) \tag{7}$$

$$\dot{\psi}_k^{desired} = \psi_k^{desired} - \psi_k^{sensor} \tag{8}$$

Using the calculated $\dot{\psi}_k^{desired}$, the heading velocity of the drone in the next state, $\dot{\psi}_{k+1}^{drone}$, is calculated, as shown in (9).

$$\dot{\psi}_{k+1}^{drone} = min\left(\left| \dot{\psi}_k^{desired} \right|, \dot{\psi}^{Max} \right) * sign\left(\dot{\psi}_k^{desired} \right), \tag{9}$$

where $\dot{\psi}^{Max}$ indicates the maximum heading rate defined by the user.

Using $\dot{\psi}_{k+1}^{drone}$, the current heading ψ_k^{drone} and the position $\vec{D}_k$ of the drone, the position $\vec{D}_{k+1}$ and velocity $\vec{V}_{k+1}^{drone}$ of the drone's next state are calculated using (10) and (11), respectively.

$$\vec{V}_{k+1}^{drone} = \left\| \vec{V}_k^{drone} \right\| \begin{bmatrix} cos\left(\psi_k^{drone} + \dot{\psi}_{k+1}^{drone} T \right) \\ sin\left(\psi_k^{drone} + \dot{\psi}_{k+1}^{drone} T \right) \end{bmatrix} \tag{10}$$

$$\vec{D}_{k+1} = \vec{D}_k + \vec{V}_{k+1}^{drone} T \tag{11}$$

$\left\| \vec{V}_k^{drone} \right\|$ represents the user-defined velocity of the drone. The simplified drone model is defined as a model flying at a constant speed.

During GNSS deception, fixed-wing drones are affected by their position, velocity, and heading, as measured by the GNSS receiver. Therefore, in Fig. 5, the directly affected position, velocity, and heading are shown in red text, and the VTP, desired velocity, and desired heading rate, which changed accordingly, are shown in blue.

Therefore, the simplified drone model during GNSS deception is shown in (12)–(15), where the position and velocity values measured from the sensors become the deceived position $\vec{D}_k^{sp}$ and velocity $\vec{V}_k^{sp}$ of the GNSS deception, and $\dot{\psi}_k^{desired}$ changes accordingly. Thus, it is possible to model the changes in the next state of the drone owing to GNSS deception.

$$\vec{V}_k^{desired} = \left(\begin{bmatrix} x_k^{VTP} \\ y_k^{VTP} \end{bmatrix} - \vec{D}_k^{sp} \right) T^{-1} \tag{12}$$

$$\psi_k^{desired} = atan2(\vec{V}_{k,y}^{desired}, \vec{V}_{k,x}^{desired}) \tag{13}$$

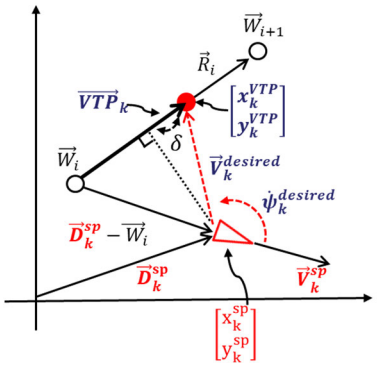

FIGURE 5. Deception effect of fixed-wing drone.

$$\psi_k^{sp} = atan2(\vec{V}_{k,y}^{sp}, \vec{V}_{k,x}^{sp}) \tag{14}$$

$$\dot{\psi}_k^{desired} = \psi_k^{desired} - \psi_k^{sp} \tag{15}$$

D. RADAR MODEL

For RADAR, the FIELDctrl range of APS in Poland was used [28] to measure the three-dimensional position and speed of the drones. RADAR modeling was performed in an RL environment. Therefore, RADAR modeling was performed based on the RADAR error measurement, which refers to the difference between the RADAR measurement data and the logging data of Remo-M flying 400 m above ground level (AGL). Seven flight tests for RADAR measurements were performed within a ground distance of 2200 m. Fig. 6 shows the measurement errors of the position and speed according to the ground distance, and is color-coded according to each flight sortie. At this point, the position error was calculated using the L2-norm of the position error of each axis, and the speed error was calculated using the absolute value of the speed difference.

Compared with other errors, the elevation error was inversely proportional to the ground distance. Because the RADAR used was installed at a height of approximately 1.5 m above the ground, it was affected by multipath ground reflection. Therefore, the elevation error, which is the slope of the error relative to the ground distance, was used to calculate the bias and bias weight. Additionally, all errors exhibited a random slope change owing to the increased prediction error of the RADAR when the drone's flight direction rapidly changed. However, we confirmed that the variance between specific sample windows was not large.

The random slope bias was set to randomly change within a range with a resolution of 10 s. The average errors of range, azimuth, elevation, and speed were used as the mean of the Gaussian distribution, respectively.

After obtaining the standard deviation of the measurement error for a specific sample window, this value was used as the standard deviation of the Gaussian distribution. As listed in Table 1, the range, azimuth, elevation, and speed errors of the RADAR were modeled by adding noise modeled by a Gaussian distribution, random slope bias, and linear regression. Fig. 7 presents the modeled RADAR noise

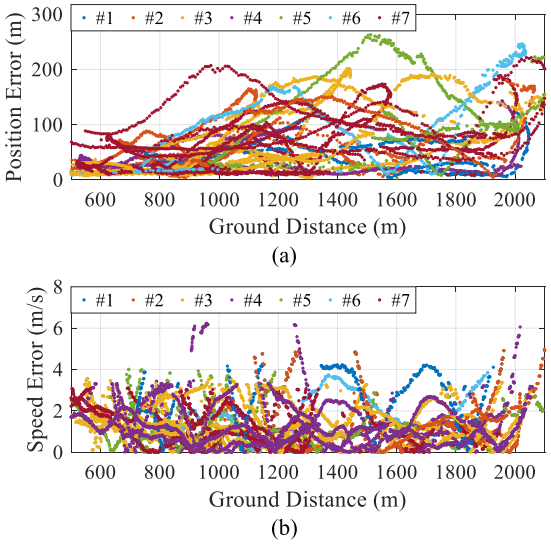

FIGURE 6. RADAR Measurements Errors. (a) Position Error. (b) Speed Error.

TABLE 1. Parameters of RADAR noise model.

Noise Type		Range (m)	Az (°)	El (°)	Speed (m/s)
Gaussian Distribution	Mean	–0.6	0.57	2.13	0
	Std.	0.5	0.1	0.1	0.1
Linear Regression	Bias Weight /Bias	-	-	–0.014 /6.5	-
Random Slope Bias	Range	–10 ~ 10	–6 ~ 6	–3.1 ~ 3.1	–3 ~ 3
	Range Resolution	5	3	1.55	1
	Changing Time	10 s	10 s	10 s	10 s

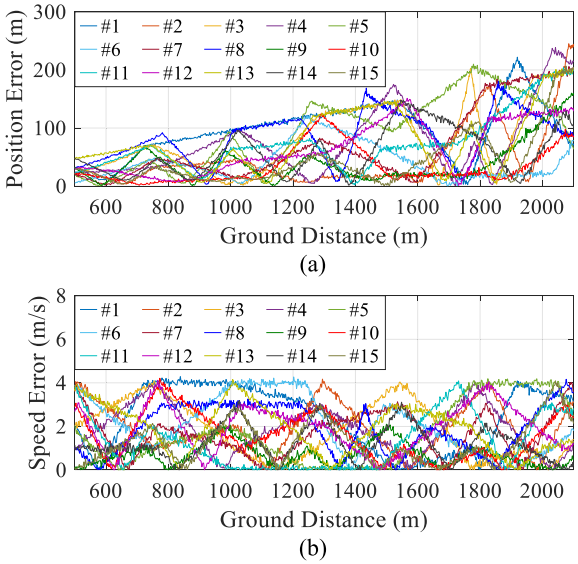

FIGURE 7. Simulation results of RADAR model. (a) Position error. (b) Speed error.

obtained through 15 simulations. The error generated by the RADAR model partially included the range of the measured error.

III. REINFORCEMENT LEARNING FOR FIXED-WING REDIRECTION

The reinforcement learning training configuration for a fixed-wing drone redirection is shown in Fig. 8. The RL agent uses proximal policy optimization (PPO), receives the state, reward, and done from the environment, and outputs the action. In this case, done was used only in the training and to determine whether the episode was terminated. The environment included the drone and RADAR models in subsections C and D of Section II, Kalman filter of the GNSS spoofer, state calculation, reward calculation, and deception position/velocity calculations. The state calculation was used to output the state using the measured drone position and velocity, which are the outputs of the Kalman filter. State s_k at the k-th time step was calculated using (16).

$$s_k = (\angle\vec{T}_k - \angle\vec{V}_k)/\pi \tag{16}$$

The target direction $\angle\vec{T}_k$ refers to the direction of the vector connecting the drone's position measured by RADAR to the designated target position, and $\angle\vec{V}_k$ represents the direction of the drone's velocity measured by RADAR. The state was normalized to a value between –1 and 1. The reward calculation uses the output of the state s_k^{bN} before normalization and the Kalman filter to calculate the reward and done status *Done*, which are calculated using (17).

The reward is set to reduce the absolute value of s_k^{bN} to match $\angle\vec{T}_k$ and $\angle\vec{V}_k$. Therefore, it is set to reward if the absolute value of s_k^{bN} is within 1° or if $\dot{s}_k^{bN}$, the difference between the absolute value of s_k^{bN} and the absolute value of the previous state s_{k-1}^{bN}, is less than zero. Additionally, if *Dist*, the distance between the designated target position and drone position, is less than or equal to r_{goal}, the episode is terminated with a large reward. A penalty is imposed when the absolute value of s_k^{bN} exceeds 90° or the first conditional statement is not satisfied.

$$r_k = \begin{cases} -0.1, & if \left|s_k^{bN}\right| > \frac{\pi}{2} \\ 0.3cos\left(s_k^{bN}\right), & elseif \left|s_k^{bN}\right| < \frac{\pi}{180} \\ 0.1cos\left(s_k^{bN}\right) + 0.2\sin(\left|\dot{s}_k^{bN}\right|), & elseif\ \dot{s}_k^{bN} < 0 \\ -0.1sin\left(\left|s_k^{bN}\right|\right) & otherwise \\ 100, Done, & if\ Dist \le r_{goal} \\ -100, Done, & elseif\ cnt > 1.5\frac{WP_{dist}}{Vel \times T} \\ 0, cnt{+}{+}, & otherwise \end{cases} \tag{17}$$

Additionally, if the number of sample times *cnt* exceeds 1.5 times the number of sample times required to reach the drone's waypoint, a large penalty is imposed, and the episode is terminated. If the second conditional statement is not met, *cnt* is increased by 1.

In actual scenarios, WP_{dist} is unknown; therefore, it was used as a variable to set *Done* to terminate an episode during training. The action output a_k by the RL agent is a variable that determines the direction of deception velocity, and values between –1 and 1 are discretely output in 201 steps. The output action was input to the drone model after the deception position/velocity calculations were performed in the environment.

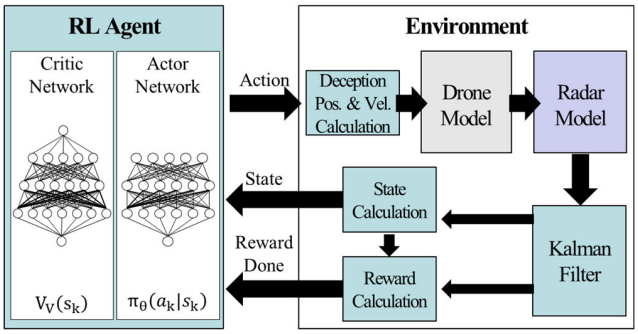

FIGURE 8. RL training configuration for fixed-wing redirection.

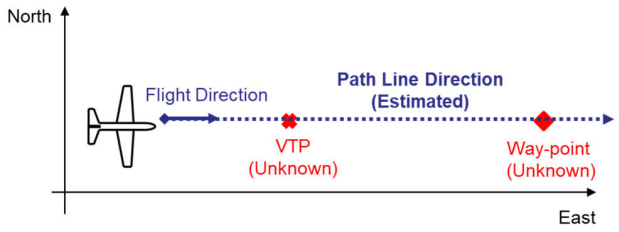

FIGURE 9. Estimation of the path line direction.

In this study, the path line direction of the fixed-wing drone before deception was estimated based on the RADAR measurement results, as shown in Fig. 9, and the deception position/velocity calculations were designed accordingly.

In [23], a method for estimating the desired heading toward a VTP was proposed. However, it operates only on the assumption that the desired heading rate does not change, even during deception. This assumption differs from (15), and it cannot be used in this study. Therefore, rather than estimating the drone's VTP or waypoint, which are difficult, a simple estimation method was used to improve the implementation.

The direction of the deception velocity is calculated using the action, the maximum angle of deception θ_{Max}, and the estimated path line direction $\hat{H}_{init}$, as shown in Fig. 10 and (18).

$$\angle\vec{V}_k^{SP} = a_k\theta_{Max} + \hat{H}_{init} \tag{18}$$

Subsequently, the deception position of the current time step is calculated using the deception position of the previous time step and the deception velocity of the current time step, as shown in (19).

$$\vec{D}_k^{SP} = \vec{D}_{k-1}^{SP} + \vec{V}_k^{SP}T \tag{19}$$

Drone modeling was also performed for two fixed-wing drones with different flight characteristics. UAV-A is a home-grown drone with a faster flight speed but a slower heading rate than Remo-M, which is a South Korean commercial fixed-wing drone. Table 2 lists the modeled parameters.

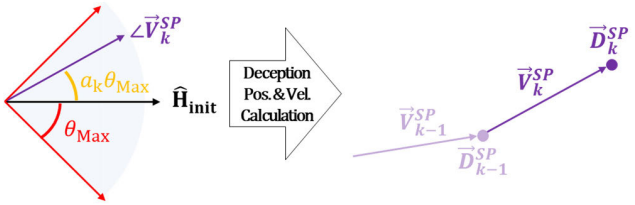

FIGURE 10. Deception position and velocity calculation in environment.

TABLE 2. Drone modeling parameters for two drones.

Parameters	UAV-A	Remo-M
$\dot{\psi}^{Max}$	7 °/s	40 °/s
$\|\vec{V}_k^{drone}\|$	27.7 m/s	16.6 m/s
WPDist	40 m	17 m
δ	60 m	20 m
ψ_0^{drone}	180 °	180 °
$\vec{D}_0$	[0, 0] m	[0, 0] m
T	0.1 s	0.1 s

TABLE 3. Training information of RL agent.

Parameters	UAV-A
Critic Network	1-512-256-128-1
Actor Network	1-256-256-201
Activation Function	ReLU (Critic)
	Tanh (Actor)
Batch Size	128
Optimizer	Adam
Learning Rate	1e-4
Discount Factor	0.95
r_{goal}	100 m
Designated Target x axis	−2500 ~ 2500 m 500 m step
Designated Target y axis	−2500 ~ 2500 m 500 m step
WP1	[0, 0] m
WP2	[10,000, 0] m

WPDist refers to the distance at which the waypoint is judged to have been reached when the distance is within *WPDist* based on the waypoint. ψ_0^{drone} and $\vec{D}_0$ indicate the initial states required for the simulation, and the same values are used for both drone models.

Table 3 lists the RL agent training parameters, waypoints, and designated target ranges.

During training, the designated target position was randomly changed for each episode. Moreover, the RL agent was first trained on UAV-A, which had the slower heading rate among the two drone models, and additional training was then performed on Remo-M. Subsequently, two drones are randomly selected for training during each episode.

The RL-based counter fixed-wing system was operated by applying the trained RL agent, the environment's state and deception position/velocity calculations to the GNSS spoofer redirection algorithm, as shown in Fig. 1.

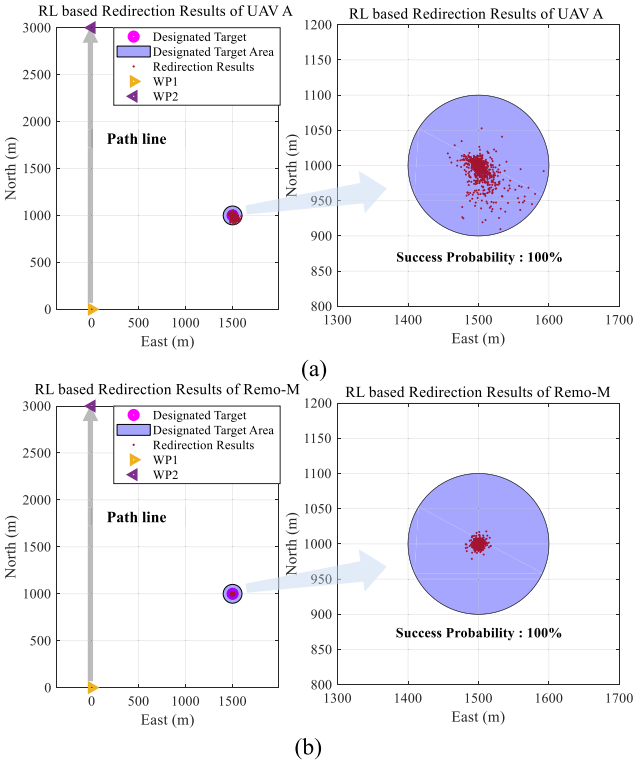

FIGURE 11. Simulation results of redirection possibility. (a) UAV-A. (b) Remo-M.

IV. SIMULATION RESULTS

The probability of successful redirection of the trained RL agent was confirmed by performing 1000 simulations. Based on the designated target position, a circle with a radius of 100 m is marked as the purple area, and the probability was calculated by checking whether the drone entered the radius. Current waypoint (WP1), next waypoint (WP2), and the designated target position were set to [0, 0], [3,000, 0], and [1000, 1500] m, respectively. As shown in Fig. 11, the simulation results showed a redirection success probability of 100% for UAV-A and Remo-M.

V. EXPERIMENTAL RESULTS

Redirection for the two types of fixed-wing drones was verified through flight tests by applying the trained RL agent to a counter fixed-wing drone system. The configuration of the RL-based counter fixed-wing drone system used for the flight testing is shown in Fig. 12.

The results of the three redirections of UAV-A and Remo-M are shown in Fig. 13 and 14, respectively. The redirection tests were performed to verify that redirection was possible even if the test conditions changed. Blue indicates the position measurement of the drone from the RADAR, red indicates the deception position, and the flight trajectory of the drone starts from the black point. The magenta triangle indicates the designated target position.

Fig. 13(b), 14(a), and 14(c) show that RADAR tracking loss occurred during redirection, but redirection to the designated target position was possible. In Fig. 13(c) and 14(b),

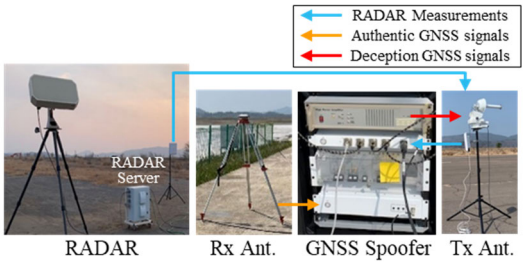

FIGURE 12. Configuration of RL-based counter fixed-wing drone system.

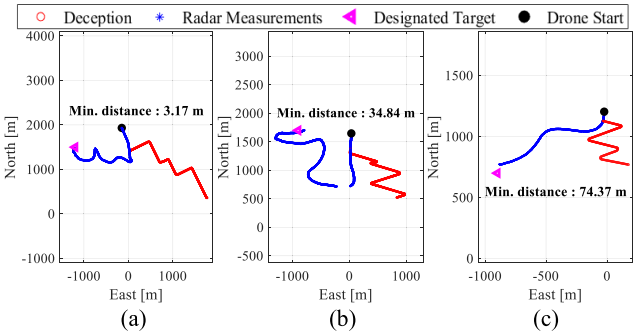

FIGURE 13. Redirection results of UAV-A based on the designated target position: (a) [1500,−1200] m. (b) [1700,−900] m. (c) [700,−900] m.

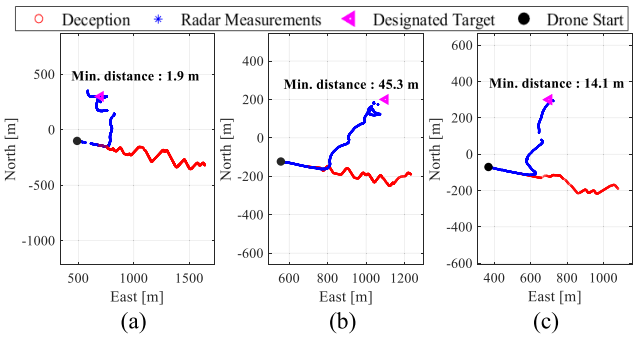

FIGURE 14. Redirection results of Remo-M based on the designated target position: (a) [300, 700] m. (b) [200, 1100] m. (c) [300, 700] m.

RADAR tracking loss occurs before reaching the designated target position, resulting in a larger redirection error than the other results. When the drone changed its flight direction rapidly, the RADAR measurement error increased, and the measured trajectory of the drone differed from the actual flight trajectory in some curved sections, as shown in Fig. 13(a) and 14(a)–(c).

Because the drone redirection tests were conducted in an area where RADAR's detection quality probability was high, RADAR tracking loss occurred owing to an increase in prediction error caused by rapid changes in the drone's position and velocity. Additionally, tracking loss occurs stochastically when the drone begins to change direction or moves in another direction because the predicted value and measurement errors follow a probability distribution.

The flight test results in Fig. 13 and 14 are all redirected within a radius of 100 m around the designated target position.

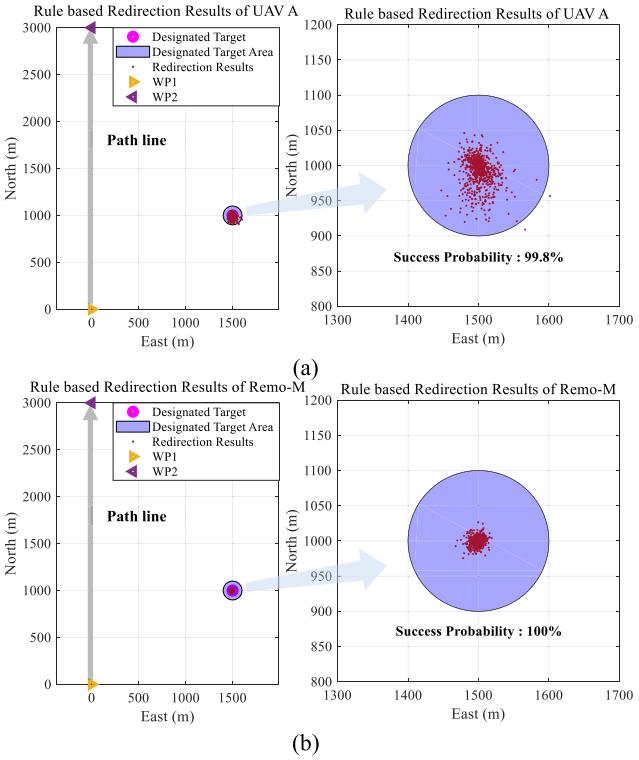

FIGURE 15. Simulation results of redirection possibility using rule-based redirection algorithm. (a) UAV-A. (b) Remo-M.

VI. DISCUSSION

We verified that an RL-based counter fixed-wing drone system trained using a simplified drone model could redirect two types of fixed-wing drones during flight testing in an untrained environment.

Fig. 15 shows the redirection probability obtained using the rule-based redirection algorithm under the same conditions as those shown in Fig. 11. The results shown in Fig. 15 exhibit a redirection success probability similar to that of RL-based redirection within a 100 m radius based on the designated target position. However, it is observed that the results of RL-based redirection are more concentrated in the designated target position. Table 4 presents a comparison of the success probabilities of the RL-based and rule-based redirection by changing the radius based on the designated target position. It can be observed that as the radius is reduced, the difference between the success probabilities of the RL-based and rule-based redirection increases.

However, compared with the flight test results of rule-based Cases 1 and 2 in [25] and Fig. 14, the rule-based results showed that the drone's flight trajectory was closer to a straight line. As a result, RADAR tracking loss occurred more frequently in the RL-based flight test results. Because a Kalman filter was used, the deception path could still be generated despite the RADAR tracking loss; however, it is necessary to generate a deception path to reduce the probability of tracking loss. This is because of the difference between the actual flight test and the simplified drone model, which is a trained environment, and it was confirmed

TABLE 4. Comparison of success probabilities of deception between RL-based and rule-based redirection.

Method	Radius	UAV-A	Remo-M
RL-based	25 m	85.3 %	100 %
	50 m	95.6 %	100 %
	100 m	100 %	100 %
Rule-based	25 m	78.5 %	99.3 %
	50 m	94.6 %	100 %
	100 m	99.8 %	100 %

that training through flight tests is also necessary for improvement.

In the training environment, RL outperformed the rule-based method; however, in a real environment, which is different from the training environment, the rule-based method performed better. This is because the rule-based performance was improved through tuning based on the flight test results, whereas the RL agent did not perform flight test-based training. Therefore, to train an RL agent in a real environment, it is necessary to perform training through online learning or, if this is limited, obtain the training data using rule-based methods during the actual operation and train through offline learning using a network capable of off-policy [29], [30].

VII. CONCLUSION

In this study, we designed and verified an RL-based counter fixed-wing drone system that can redirect a fixed-wing drone to a designated target position during automatic flight. For RL training, a simplified drone model, RADAR model, GNSS spoofer's Kalman filter, state calculation, reward calculation, and deception position/velocity calculations were used to construct an environment. The measurement error of the RADAR model was modeled based on the measurement results of the RADAR using the flight test results of the fixed-wing drone. Subsequently, the reference angle of action for the fixed-wing drone redirection was set by estimating the path line direction, and the MDP was defined accordingly. The redirection success probability of the designed RL-based counter fixed-wing drone system was confirmed through a simulation in a trained environment for two types of fixed-wing drones, and the redirection was confirmed in flight tests for two types of fixed-wing drones in an untrained environment. Therefore, the designed RL-based counter fixed-wing drone system can redirect the fixed-wing drone to the designated target position without any prior information, such as the waypoint or VTP of the drone.

Additionally, the redirection results of the RL agent were compared with the rule-based results in both trained and untrained environments. The RL agent redirection in the untrained environment produced a rapid change in the drone flight direction when compared to that of the rule-based results, which frequently caused RADAR tracking loss. Therefore, the training of RL agents needs to be studied based on drone flight tests using online or offline learning to improve the performance of RL-based redirection in drone flight tests.

In this study, we verified a method for training the RL agent of the RL-based counter fixed-wing drone system using simplified fixed-wing drone models, and the designed system is expected to be applied to anti-drone systems to counter fixed-wing drones.

REFERENCES

[1] S. O. Ajakwe, D.-S. Kim, and J. M. Lee, "Drone transportation system: Systematic review of security dynamics for smart mobility," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14462–14482, Aug. 2023, doi: 10.1109/JIOT.2023.3266843.

[2] M. R. DeVore, "'No end of a lesson': Observations from the first high-intensity drone war," *Defense Secur. Anal.*, vol. 39, no. 2, pp. 263–266, Apr. 2023, doi: 10.1080/14751798.2023.2178571.

[3] C. Lyu and R. Zhan, "Global analysis of active defense technologies for unmanned aerial vehicle," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 37, no. 1, pp. 6–31, Jan. 2022, doi: 10.1109/MAES.2021.3115205.

[4] H. Kang, J. Joung, J. Kim, J. Kang, and Y. S. Cho, "Protect your sky: A survey of counter unmanned aerial vehicle systems," *IEEE Access*, vol. 8, pp. 168671–168710, 2020, doi: 10.1109/ACCESS.2020.3023473.

[5] D. Schmidt, K. Radke, S. Camtepe, E. Foo, and M. Ren, "A survey and analysis of the GNSS spoofing threat and countermeasures," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 1–31, May 2016, doi: 10.1145/2897166.

[6] D. He, G. Yang, H. Li, S. Chan, Y. Cheng, and N. Guizani, "An effective countermeasure against UAV swarm attack," *IEEE Netw.*, vol. 35, no. 1, pp. 380–385, Jan. 2021, doi: 10.1109/MNET.011.2000380.

[7] S. Park, H. T. Kim, S. Lee, H. Joo, and H. Kim, "Survey on anti-drone systems: Components, designs, and challenges," *IEEE Access*, vol. 9, pp. 42635–42659, 2021, doi: 10.1109/ACCESS.2021.3065926.

[8] Y. N. Jurn, S. A. Mahmood, and J. A. Aldhaibani, "Anti-drone system based different technologies: Architecture, threats and challenges," in *Proc. 11th IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Penang, Malaysia, Aug. 2021, pp. 114–119, doi: 10.1109/ICCSCE52189.2021.9530992.

[9] Z. Renyu, S. C. Kiat, W. Kai, and Z. Heng, "Spoofing attack of drone," in *Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, Chengdu, China, Dec. 2018, pp. 1239–1246, doi: 10.1109/COMPCOMM.2018.8780865.

[10] S. Daniel, B. A. Jashan, and H. E. Todd, "Drone hack: Deception attack demonstration on a civilian unmanned aerial vehicle," GPS World, Cleveland, OH, USA, Tech. Rep., 2012, pp. 30–33.

[11] S. P. Arteaga, L. A. M. Hernández, G. S. Pérez, A. L. S. Orozco, and L. J. G. Villalba, "Analysis of the GPS spoofing vulnerability in the drone 3DR solo," *IEEE Access*, vol. 7, pp. 51782–51789, 2019, doi: 10.1109/ACCESS.2019.2911526.

[12] M. L. Psiaki and T. E. Humphreys, "GNSS spoofing and detection," *Proc. IEEE*, vol. 104, no. 6, pp. 1258–1270, Jun. 2016, doi: 10.1109/JPROC.2016.2526658.

[13] D. Mendes, N. Ivaki, and H. Madeira, "Effects of GPS spoofing on unmanned aerial vehicles," in *Proc. IEEE 23rd Pacific Rim Int. Symp. Depend. Comput. (PRDC)*, Taipei, Taiwan, Dec. 2018, pp. 155–160, doi: 10.1109/PRDC.2018.00026.

[14] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, "GPS vulnerability to spoofing threats and a review of antispoofing techniques," *Int. J. Navigat. Observ.*, vol. 2012, pp. 1–16, Jul. 2012, doi: 10.1155/2012/127072.

[15] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS spoofing," *J. Field Robot.*, vol. 31, no. 4, pp. 617–636, Jul. 2014, doi: 10.1002/rob.21513.

[16] Y. Guo, M. Wu, K. Tang, J. Tie, and X. Li, "Covert spoofing algorithm of UAV based on GPS/INS-integrated navigation," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6557–6564, Jul. 2019, doi: 10.1109/TVT.2019.2914477.

[17] J. Gaspar, R. Ferreira, P. Sebastião, and N. Souto, "Capture of UAVs through GPS spoofing," in *Proc. Global Wireless Summit (GWS)*, Chiang Rai, Thailand, Nov. 2018, pp. 21–26, doi: 10.1109/GWS.2018.8686727.

[18] J. Noh, Y. Kwon, Y. Son, H. Shin, D. Kim, J. Choi, and Y. Kim, "Tractor beam: Safe-hijacking of consumer drones with adaptive GPS spoofing," *ACM Trans. Privacy Secur.*, vol. 22, no. 2, pp. 1–26, May 2019, doi: 10.1145/3309735.

[19] R. Ferreira, J. Gaspar, P. Sebastião, and N. Souto, "A software defined radio based anti-UAV mobile system with jamming and spoofing capabilities," *Sensors*, vol. 22, no. 4, p. 1487, Feb. 2022, doi: 10.3390/s22041487.
[20] H. Sathaye, M. Strohmeier, V. Lenders, and A. Ranganathan, "An experimental study of GPS spoofing and takeover attacks on UAVs," in *Proc. USENIX Secur. Symp.*, 2022, pp. 3503–3520.
[21] D. He, Y. Qiao, S. Chen, X. Du, W. Chen, S. Zhu, and M. Guizani, "A friendly and low-cost technique for capturing non-cooperative civilian unmanned aerial vehicles," *IEEE Netw.*, vol. 33, no. 2, pp. 146–151, Mar. 2019, doi: 10.1109/MNET.2018.1800065.
[22] W. Chen, Y. Dong, and Z. Duan, "Accurately redirecting a malicious drone," in *Proc. IEEE 19th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2022, pp. 827–834, doi: 10.1109/CCNC49033.2022.9700664.
[23] C. Ma, J. Yang, J. Chen, and C. Zhou, "Path following identification of unmanned aerial vehicles for navigation spoofing and its application," *ISA Trans.*, vol. 108, pp. 393–405, Feb. 2021, doi: 10.1016/j.isatra.2020.08.016.
[24] M. Li, Y. Kou, Y. Xu, and Y. Liu, "Design and field test of a GPS spoofer for UAV trajectory manipulation," in *Proc. China Satell. Navigat. Conf.*, in Lecture Notes in Electrical Engineering, 2018, pp. 161–173, doi: 10.1007/978-981-13-0014-1_15.
[25] M.-H. Chae, S.-O. Park, S.-H. Choi, and C.-T. Choi, "Commercial fixed-wing drone redirection system using GNSS deception," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 5, pp. 5699–5713, Oct. 2023, doi: 10.1109/TAES.2023.3264193.
[26] D. L. D. Silva, R. Machado, O. L. Coutinho, and F. Antreich, "A soft-kill reinforcement learning counter unmanned aerial system (C-UAS) with accelerated training," *IEEE Access*, vol. 11, pp. 31496–31507, 2023, doi: 10.1109/ACCESS.2023.3253481.
[27] G. G. Yen and T. W. Hickey, "Reinforcement learning algorithms for robotic navigation in dynamic environments," *ISA Trans.*, vol. 43, no. 2, pp. 217–230, Apr. 2004, doi: 10.1016/s0019-0578(07)60032-9.
[28] *APS FIELDctrl Range 3D Radar*. [Online]. Available: https://apsystems.tech/en/products/ultra-precise-3d-mimo-radars
[29] J. Ge, Y.-C. Liang, J. Joung, and S. Sun, "Deep reinforcement learning for distributed dynamic MISO downlink-beamforming coordination," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6070–6085, Oct. 2020, doi: 10.1109/TCOMM.2020.3004524.
[30] R. F. Prudencio, M. R. O. A. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE Trans. Neural Netw. Learn. Syst.*, 2023, doi: 10.1109/TNNLS.2023.3250269.

MYOUNG-HO CHAE received the B.S. and M.S. degrees in electrical engineering from Chungnam National University, South Korea, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with the Korea Advanced Institute of Science and Technology, Daejeon, South Korea. He is also a Senior Researcher with Agency for Defense Development, Daejeon. His research interests include wideband frequency synthesizer, wideband receiver, and electronic warfare systems.

SEONG-OOK PARK (Senior Member, IEEE) was born in Kyungpook, South Korea, in December 1964. He received the B.S. degree in electrical engineering from Kyungpook National University, South Korea, in 1987, the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1989, and the Ph.D. degree in electrical engineering from Arizona State University, Tempe, in 1997. From March 1989 to August 1993, he was a Research Engineer with Korea Telecom, Daejeon, working with microwave systems and networks. Later, he joined the Telecommunications Research Center, Arizona State University and worked there until September 1997. Since October 1997, he has been with Information and Communications University, Daejeon. He is currently a Professor with the Korea Advanced Institute of Science and Technology. His research interests include mobile handset antennas and analytical and numerical techniques in electromagnetics. He is a member of the Phi Kappa Phi.

SEUNG-HO CHOI received the B.S. degree in electrical engineering from Yeung Nam University, South Korea, in 1992, the M.S. degree in electrical engineering from the Pohang University of Science and Technology, in 1998, and the Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 2008. Later, he joined Agency for Defense Development, Daejeon, where he is currently a Principal Researcher. His research interest includes electronic attack technologies for EW.

CHAE-TAEK CHOI received the B.S. and M.S. degrees in computer science and statistics from Chungnam National University, South Korea, in 1989 and 1991, respectively. He has been a Principal Researcher with Agency for Defense Development, since 1991. His research interests include neural network optimization, RF-counter unmanned aerial system technology, and electronic warfare systems.