Commercial Fixed-Wing Drone Redirection System using GNSS Deception

Myoung-Ho Chae

Korea Advanced Institute of Science and Technology, Daejeon, Korea
Agency for Defense Development, Daejeon, Korea

Seong-Ook Park, *Member, IEEE*

Korea Advanced Institute of Science and Technology, Daejeon, Korea

Seung-Ho Choi

Chae-Taek Choi

Agency for Defense Development, Daejeon, Korea

***Abstract*—As drones become more common today, the threat of reconnaissance or attack drones to core facilities has increased, and countermeasures against them have become essential. In this study, a drone redirection system was proposed to counter illegal intrusion of commercial fixed-wing drones. The drone redirection system was designed as a closed-loop system that automatically redirects the drone to a target position. The main novelty of this study is a proposal for a system that can automatically redirect a commercial fixed-wing drone, which has not been previously explored. We proposed two strategies for redirecting drones. Additionally, simple drone modeling with a path-following algorithm was used to easily model various drones. The drone model was then tuned using flight test data, and the results were compared. Simulations were performed on the designed drone redirection system model to verify the performance of the two proposed strategies for redirecting drones in conjunction with drone fail-detection and innovation check. The performance of the drone redirection system was assessed through flight tests of Remo-M and simulations of Micropilot's hardware-in-the-loop simulator (HWILS). Through simple drone modeling, drone flight tests, and the test results from HWILS, it was proved that the drone redirection system can be applied to various fixed-wing drones.**

***Index Terms*—Anti-drone System, Closed loop systems, Electronic countermeasures, GNSS deception.**

Manuscript received June 3, 2019; revised October 26, 2019, and January 7, 2020; released for publication February 17, 2020.

DOI. No. 10.1109/TAES.2020.2977792

This work was supported in part by the Agency for Defense Development (ADD) (912759101) ***(Corresponding author: Myoung-Ho Chae.)***

M. Chae is with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea, and also with the Agency for Defense Development, Daejeon 34186, South Korea, e-mail:(mhchae@kaist.ac.kr); S. Park is with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea, e-mail:(soparky@kaist.ac.kr); S. Choi and C. Choi are with Agency for Defense Development, Daejeon 34186, South Korea, e-mail:(sh_choi@add.re.kr; ctmarine@naver.com).

I. INTRODUCTION

Owing to the increasing use of drones in recent years, cases of illegal drones invading core facilities have significantly increased [1–3]. Therefore, several studies are being conducted to develop methods to counter illegal drones. These methods can be divided into hard-kill, which inflict physical damage, and soft-kill, which can suppress without physical damage. Soft-kill methods include jamming and deception of the global navigation satellite system (GNSS), and remote-control signals of drones through electromagnetic wave transmission.

Although GNSS deception can counter illegal drones with low power compared to that used in jamming, deception is possible only on civilian GNSS signals [4–6]. GNSS deception deceives the target GNSS receiver in a way that the time delay of the signal of each satellite is changed to intentionally change the position, velocity, and time (PVT) of the target GNSS receiver [7–10]. This can be implemented by manipulating the navigation message [11] or using code delay and Doppler change [12].

Several studies have analyzed the effects of GNSS deception. A study was conducted on deviating a ship from its route using GNSS deception that targeted large ships [13]. Another study analyzed the responses of multi-rotors by various GNSS deception attack models using simulation-in-the-loop (SITL) on open-source software [14].

Studies have also been conducted on moving drones to safe areas to prevent them from moving in the wrong direction or to avoid accidents in unintended places during GNSS deception. In [15,16], closed-loop simulations of drone and deception models were performed in a two-dimensional horizontal plane, confirming that the drone followed the desired path as the attack reference. However, the target acceleration of the drone model was pre-planned, regardless of the deception position and velocity. Thus, the result was not a realistic representation of flight algorithm of a real drone. In [15], deception was performed using a small unmanned helicopter to analyze the effect of deception on the extended Kalman filter (EKF) of the helicopter. In [17], a concept for redirecting drones to a landing area using deception signals generated using software-defined radio (SDR) was presented, and the results of deception tests on GNSS receivers and smartphones were demonstrated. In [18], a path-following algorithm for multi-rotors was analyzed, and hard deception was performed using an open-loop without drone detection, thereby demonstrating the direction change errors of DJI phantoms 3 and 4, 3DR solo, and parrot bebop2 as simulation results of open-source SITL or experiments. Reference [19] confirmed through simulation that the multi-rotor parrot model of AR Drone 2.0 can be redirected to a target position using a redirection algorithm for a closed-loop structure. In addition, the possibility of the manual control of a hovering multi-rotor using GNSS deception in a human-in-the-loop (HITL) configuration was demonstrated through

experiments. One study was conducted on moving a multi-rotor in a desired direction using a gun-type GNSS spoofer [20], whereas another study confirmed the results of the redirection of a multi-rotor to a target position using open-source SITL [21]. However, because the presented algorithm needs to know the way-point of the multi-rotor in advance, it is difficult to use this method in real situations. Similar to [19], a study confirmed the controllability of a multi-rotor using GNSS deception [22]. This suggests, based on simulation results, that more stable control of a hovering multi-rotor can be achieved by controlling the deception speed according to the speed of the drone.

Although, studies targeting multi-rotor drones have already been conducted, to the best of our knowledge, no studies targeting fixed-wing drones have been performed. Multi-rotors are easy to take off and land, and can hover, but have limited flight time capability and coverage area. By contrast, fixed-wing drones have longer flight times and flight distances [23]. Therefore, countermeasures against fixed-wing drones are highly necessary. Because of the differences in the flight characteristics of fixed-wing and multi-rotor drones, it is not possible to redirect illegally intruding fixed-wing drones using existing research results.

In addition, most previous studies conducted without drone detection devices, analyzed responses to the GNSS deception of multi-rotor drones with an open-loop, and established strategies to reach a target position or direction. By comparison, when the redirection algorithm is configured in a closed-loop structure, the redirection accuracy and response speed are higher than those in an open-loop or HITL, and the user convenience of responding to a drone is high. However, no previous studies have thus far been found verifying the results of redirection using a closed-loop through an experiment in a real environment, whether with fixed-wing or multi-rotor drones. For experimental verification in a real environment, an algorithm using a closed-loop should be designed, considering the performance of the drone detection device.

In this study, we conducted a GNSS deception study on redirecting an illegally intruding fixed-wing drone to reach a target position, while targeting the fixed-wing drone in auto-flight using GNSS. The contributions of this study are as follows:

(1) Two strategies for redirecting a fixed-wing drone to a target position were proposed and applied in a drone redirection algorithm and system model, the results of which confirmed that redirection to a target position is possible in simulations of simple fixed-wing drone modeling using various path-following algorithms.
(2) The drone redirection system, which is composed of a closed-loop that uses RADAR, validated that redirection to a target position is possible, based on a commercial fixed-wing flight test.
(3) To confirm that the drone redirection algorithm can be applied to various fixed-wing drones, the algorithm was verified through a simulation using MicroPilot's hardware-in-the-loop-simulator (HWILS).

The remainder of this paper is organized as follows. Section II presents the modeling method and its results based on the path-following algorithm of the fixed-wing drone. Section III shows the proposed redirection strategies and system configuration. Section IV presents the flight test and HWILS simulation results, and Section V presents a discussion of these results. Finally, Section VI presents the conclusions of this study.

II. FIXED-WING DRONE MODELING

To design an algorithm that redirects an auto-flight drone to a target position using a GNSS deception signal, it is necessary to identify the drone's tendency to change its position, as well as the velocity when applying a GNSS deception with a specific position and velocity. Then, a drone redirection algorithm can be designed to change the position and velocity of the GNSS deception signal. At this point, it is possible to effectively design a drone redirection algorithm through simulations using drone modeling.

There are several simulators available from open-source SITL and autopilot companies [24]. To design the drone redirection algorithm using these simulators, it is necessary to tune the drone model of the simulators similarly to the actual flight result of the drone. To tune the drone model for the simulation, the aircraft size, aerodynamic, and thrust coefficient, which are the aircraft information of the drone, are input; the type of the path-following algorithm is selected; the controller structure is tuned, and the control variables and other variables are set in detail. However, except for drones that apply autopilot using open-source code, the information for drone tuning, reference sensor setting for flight control, sensor fusion conditions, and fail-detection conditions are not disclosed. Furthermore, to determine the detailed drone information for drone tuning, it is necessary to use a reverse engineering approach for the target drone.

In this study, the most commonly used path-following algorithms, sensor fusion conditions, and fail-detection conditions were investigated, referring to published literature and open-sources, and the drone was modeled in a simple manner. Several tuning variables can be reduced through simple drone modeling; it was possible to tune a simple drone model within a relatively short time based on the flight test results for the target drone. With a simple drone model, the similarity is found to be less than that of a detailed drone model; however, it can be easily implemented for drones with various flight characteristics, fail-detection, and fusion conditions. Therefore, it is possible to easily establish a concept for a drone redirection algorithm at the initial design stage and design a general-purpose drone redirection algorithm. In [25], in a similar manner, an unmanned surface vehicle (USV) was modeled in a simple way to analyze the response of the USV in GNSS deception. In this study, modeling was performed on a fixed-wing drone. Carrot chasing, nonlinear guidance law (NLGL), and vector field (VF), which are the mainly used path-following algorithms among those published in various literatures and open sources [26–28], were modeled. Then, for simple modeling, the inner loop controller was omitted, and the flight speed was kept constant. Additionally, only a two-dimensional horizontal plane was modeled. Moreover, the reference sensor for control in a simplified drone model can be changed, and a fusion of GNSS and inertial measurement unit (IMU), which are generally used in drones, was modeled. It can be used to model fail-detection and innovation check.

A. Path-Following Algorithms

The path-following algorithm, whose main principle is to generate lateral acceleration to compensate for cross-track error, which is the vertical distance between the path line and drone position, places a virtual target point (VTP) on the path line to ensure the drone follows the path line. The magnitude and direction of the cross-track error are proportional to the magnitude and direction of the lateral acceleration. In the case of carrot chasing and NLGL, the VTP is set on the path line depending on the cross-track error, and a lateral acceleration is generated accordingly. In the case of VF, the vector field is designated depending on the cross-track error, and a lateral acceleration is generated accordingly. As shown in Fig. 1, carrot chasing calculates the cross-track error based on the drone position and sets VTP to the point along the path line added by δ at the intersection of the path line and cross-track error. $\overrightarrow{W_{i+1}}$ and $\overrightarrow{W_i}$ are the vectors of the i+1th and i-th way-points, respectively. $\overrightarrow{VTP_k}$ is a position vector of the VTP, and $[x_k^{VTP}, y_k^{VTP}]^T$ are position coordinates of the VTP. $[x_k^{sensor}, y_k^{sensor}]^T$ are the coordinates of the position measured from the sensor of the drone, and $\vec{D}_k^{sensor}$ is the position vector measured from the sensor. $\vec{R}_i$ is the path line vector connecting $\overrightarrow{W_{i+1}}$ and $\overrightarrow{W_i}$, and $\vec{D}_k$ is the position vector of the drone. The variable δ refers to the distance at which VTP is set. $\vec{V}_k^{desired}$ is the desired velocity vector, $\dot{\psi}_k^{desired}$ is the desired heading rate, and $\vec{V}_k^{sensor}$ is the velocity measured from the sensor of the drone. Subscript k of the variable indicates the k-th time step.

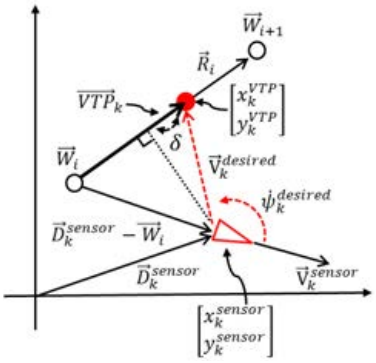

Fig. 1. Conceptual diagram of carrot chasing.

The VTP of carrot chasing can be calculated using (1)–(3).

$$\vec{R}_i = \overrightarrow{W_{i+1}} - \overrightarrow{W_i} \tag{1}$$

$$\overrightarrow{VTP_k} = \{\vec{R}_i \cdot (\vec{D}_k^{sensor} - \overrightarrow{W_i})\}\frac{\vec{R}_i}{\|\vec{R}_i\|^2} + \delta\frac{\vec{R}_i}{\|\vec{R}_i\|} \tag{2}$$

$$\begin{bmatrix} x_k^{VTP} \\ y_k^{VTP} \end{bmatrix} = \overrightarrow{VTP_k} + \overrightarrow{W_i} \tag{3}$$

By setting the variable δ, the distance between the drone and VTP can be set to adjust the degree of following the path line.

NLGL places the contact point of radius L of the circle and the path line as VTP at the position of the drone, similar to that in algorithm 4 in [26]. Therefore, in NLGL, the VTP is determined depending on the variable L; if L is small, VTP is set close to the drone, thereby increasing the degree of following the path line. Furthermore, if there is no contact point between the circle of variable L and the path line, the following angle can be set as a variable. For open-source software Ardupilot and PX4, an L1 controller based on NLGL is used [26–28]. This is a linearization of the equation for the lateral acceleration into the proportional and differential (PD) relational equations of the cross-track error.

The VF stores the desired velocity vector toward the path line as a field. The variables of VF comprise transition boundary τ, which indicates the threshold of the cross-track error; entry heading χ^e, which can set the maximum following angle of the velocity vector; and gain α [26,29–32]. As the cross-track error increases, the direction of the desired velocity vector is set closer to χ^e, whereas the direction's magnitude of the desired velocity vector is set closer to the direction of the path line as the cross-track error decreases. Additionally, when the cross-track error exceeds τ, the direction of the desired velocity vector becomes χ^e. If α is a large value, the convergence time is reduced when the drone is loitering [26].

In the case of carrot chasing and NLGL, considering VTP is calculated, $\vec{V}_k^{desired}$ and the desired heading $\psi_k^{desired}$ can be obtained using (4) and (5). In the case of VF, the direction of each vector field becomes the desired heading. $\vec{V}_{k,x}^{desired}$ and $\vec{V}_{k,y}^{desired}$, which are the x- and y-axes components of $\vec{V}_k^{desired}$.

$$\vec{V}_k^{desired} = \left(\begin{bmatrix} x_k^{VTP} \\ y_k^{VTP} \end{bmatrix} - \vec{D}_k^{sensor}\right)T^{-1} \tag{4}$$

$$\psi_k^{desired} = \text{atan2}(\vec{V}_{k,y}^{desired}, \vec{V}_{k,x}^{desired}) \tag{5}$$

As shown in (6), the heading ψ_k^{sensor} measured from the drone's sensor can be used to obtain $\vec{V}_{k,x}^{sensor}$ and $\vec{V}_{k,y}^{sensor}$, which are the x- and y-axes components of velocity $\vec{V}_k^{sensor}$, and the desired heading rate $\dot{\psi}_k^{desired}$ can be calculated as shown in (7).

$$\psi_k^{sensor} = \text{atan2}(\vec{V}_{k,y}^{sensor}, \vec{V}_{k,x}^{sensor}) \tag{6}$$

$$\dot{\psi}_k^{desired} = \psi_k^{desired} - \psi_k^{sensor} \tag{7}$$

The drone's heading rate, $\dot{\psi}_{k+1}^{drone}$, is the same as that in (8), and the variable $\dot{\psi}^{Max}$ is the maximum heading rate.

$$\dot{\psi}_{k+1}^{drone} = \min\left(\left|\dot{\psi}_k^{desired}\right|, \dot{\psi}^{Max}\right) * \text{sign}(\dot{\psi}_k^{desired}) \tag{8}$$

In this study, instead of calculating the lateral acceleration, we calculated the velocity $\vec{V}_{k+1}^{drone}$ and position $\vec{D}_{k+1}$ of the next state of the drone in a two-dimensional horizontal plane using (9) and (10).

$$\vec{V}_{k+1}^{drone} = \left\|\vec{V}_k^{drone}\right\| \begin{bmatrix} \cos\left(\psi_k^{drone} + \dot{\psi}_{k+1}^{drone}T\right) \\ \sin\left(\psi_k^{drone} + \dot{\psi}_{k+1}^{drone}T\right) \end{bmatrix}, \tag{9}$$

$$\vec{D}_{k+1} = \vec{D}_k + \vec{V}_{k+1}^{drone}T, \tag{10}$$

where ψ_k^{drone} is the drone heading, T is the sample time and $\left\|\vec{V}_k^{drone}\right\|$ is the L2-norm of the drone velocity in a two-dimensional horizontal plane, which is a variable that can set the flight speed of the drone. Therefore, the velocity and position vectors of the drone at every step can be obtained by setting the variable values of $\left\|\vec{V}_k^{drone}\right\|$ and $\dot{\psi}^{Max}$. Moreover, ψ_0^{drone} and $\vec{D}_0$, which are the initial values of ψ_k^{drone} and $\vec{D}_k$, can be set for simulation.

B. GNSS&IMU Fusion

The drone estimates its state, such as the attitude, acceleration, velocity, and position, based on a fusion of GNSS and IMU. Then, a correction is performed using the measurement data of the GNSS. The GNSS&IMU fusion

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3264193

implemented in this study is an indirect feedback structure. As shown in Fig. 2, the navigation solution is obtained using the measured value of IMU to estimate the position and velocity of the state.

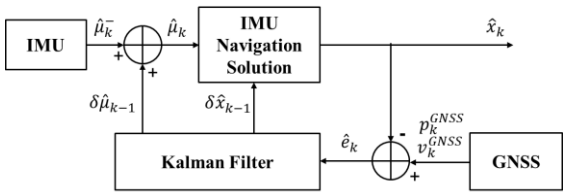

Fig. 2. Indirect GNSS&IMU fusion.

The error between the states estimated by the IMU and GNSS measurements is input to the Kalman filter to obtain the estimated correction values for the IMU and IMU navigation solution, which in turn is used for correction when calculating the IMU navigation solution in the next step. Herein, only the two-dimensional horizontal plane was considered as the state for the simplification of the model, whereas the IMU outputs only the acceleration as the navigation axis. The IMU navigation solution is the same as that in (11) and (12), and the current state is estimated using the measured IMU value and the estimated correction value of the Kalman filter.

$$\hat{\mu}_k = \hat{\mu}_k^- + \delta\hat{\mu}_{k-1}, \tag{11}$$

$$\hat{x}_k = \begin{bmatrix} \hat{p}_k \\ \hat{v}_k \end{bmatrix} = A\left(\hat{x}_{k-1} + \delta\hat{x}_{k-1}\right) + B\hat{\mu}_k, \tag{12}$$

where $\hat{\mu}_k$ and $\hat{\mu}_k^-$ are the corrected and measured accelerations of the IMU, respectively, and $\delta\hat{\mu}_{k-1}$ is the estimated acceleration correction value of the IMU in the previous step. $\hat{x}_k$ refers to the estimated state of the position $\hat{p}_k$ and velocity $\hat{v}_k$; $\delta\hat{x}_{k-1}$ is the estimated state correction value of the IMU navigation in the previous step; and A and B are the state transition matrices for the position and velocity, and acceleration, respectively. Equations (13)–(17) show the Kalman filter's equations. The Kalman filter receives an error between the measured and estimated values and outputs an estimated correction value.

$$\hat{e}_k = \begin{bmatrix} p_k^{GNSS} \\ v_k^{GNSS} \end{bmatrix} - \begin{bmatrix} \hat{p}_k \\ \hat{v}_k \end{bmatrix} \tag{13}$$

$$\hat{P}_k^- = F\hat{P}_{k-1}F^T + GQ_{k-1}G^T \tag{14}$$

$$K_k = \hat{P}_k^- H^T\left(H\hat{P}_k^- H^T + R_k\right)^{-1} \tag{15}$$

$$\hat{P}_k = \left(I - K_k H\right)\hat{P}_k^- \tag{16}$$

$$\begin{bmatrix} \delta\hat{x}_k \\ \delta\hat{\mu}_k \end{bmatrix} = \begin{bmatrix} O \\ \delta\mu_k \end{bmatrix} + K_k\,\hat{e}_k \tag{17}$$

Here, p_k^{GNSS} is the position measurement of GNSS, v_k^{GNSS} is the velocity measurement of GNSS, and $\hat{e}_k$ is the innovation, that is, the difference between the GNSS measurement and state estimated using the IMU navigation solution. $\hat{P}_k^-$ is the error covariance prediction matrix, and $\hat{P}_k$ is the error covariance matrix. F is the state transition matrix for position, velocity, and acceleration, Q_{k-1} is the noise variance matrix at the previous step, and G is the noise transfer matrix. K_k is the Kalman gain, H is the measurement matrix, and R_k is the measurement noise variance matrix. $\delta\mu_k$ is the acceleration bias error correction value of IMU, I is the identity matrix, and O is the zero matrix.

C.Sensor Selection

The mounted sensor on the drone obtains position, velocity, attitude, angular velocity, and altitude measurements. Simultaneously, whether the flight path of the drone is changed or not is determined when the GNSS deception signal is received, depending on the reference sensor used for the controller among the mounted sensors. Therefore, the drone model was modeled such that the reference sensor can be changed. In the case of heading control, some of the values measured by the IMU, magnetometer, compass, and GNSS were used individually or in combination to obtain the reference sensor value. Additionally, for position and velocity control, the values measured in GNSS and IMU can be used individually or in combination. To control the flight speed, the airspeed of the pitot tube or the speed of the GNSS is measured and used.

The sensor values used for drone modeling can be used to set ψ_k^{sensor}, $\vec{V}_k^{sensor}$, and $\vec{D}_k^{sensor}$ variables in (4) and (6). Some values measured by the IMU, magnetometer, compass, or GNSS can be used alone or in combination to set the variable ψ_k^{sensor}. Additionally, the results of GNSS or GNSS&IMU fusion or dead reckoning of IMU can be used to set variables $\vec{V}_k^{sensor}$ and $\vec{D}_k^{sensor}$. The result of GNSS or GNSS&IMU fusion, or airspeed, can be used to set the measured speed variable $\left\|\vec{V}_k^{sensor}\right\|$.

D.Simulation & Test Result

Fig. 3 shows Remo-M, a commercial fixed-wing drone, in South Korea. The flight test was performed using this drone in automatic flight mode.

Fig. 3. Remo-M: a Korean commercial fixed-wing drone

Variables tuning of the drone model was performed to ensure the flight and simulation trajectories of the flight data and drone model, respectively, were similar. The common variables of the drone model comprised $\dot{\psi}^{Max}$, $\left\|\vec{V}_k^{drone}\right\|$, and *Way-Point Distance*, a variable with a distance value that determines whether or not a drone has reached the way-point. Moreover, if the drone is located within the distance value of the *Way-Point Distance* depending on the target way-point, the drone sets the next way-point to follow the path line. $\dot{\psi}^{Max}$ was set to 40 °/s, $\left\|\vec{V}_k^{drone}\right\|$ to 16.6 m/s, and *Way-Point Distance* to 17 m. Additionally, δ of carrot chasing, L of NLGL, and τ of VF were set to 20 m, and the following angle was set to 45 ° when there was no contact point between the circle and path line drawn in L of NLGL. χ^e of VF was set to 45 °, and α was set to 10. ψ_0^{drone} was set to 180 °, and $\vec{D}_0$ was set to origin.

To compare the simple drone model and the detailed one, we also tuned the HWILS, which is Micropilot's trueHWIL^2 from Canada [33]. The aircraft size, weight, thrust coefficient, and controller parameters for the HWILS were adjusted to the trajectory of the flight data.

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3264193

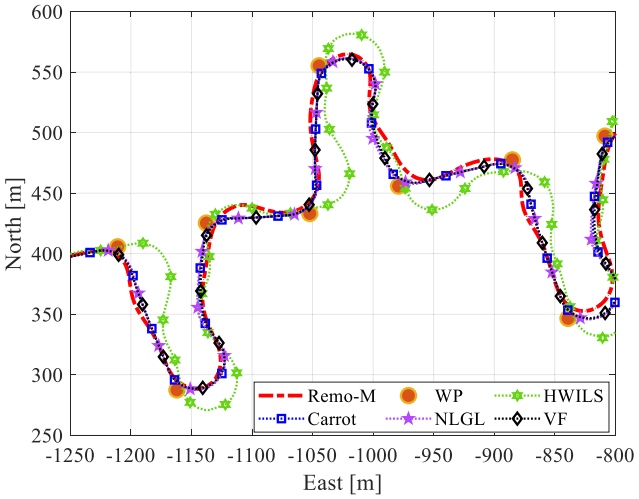

Fig. 4. Trajectory comparison: Drone modeling result, flight test result, and HWILS result.

Fig. 4 shows the modeling results, flight results of the Remo-M, and the results of the HWILS. The paths of carrot chasing, NLGL, and VF followed the way-point and path line well, and were similar to the trajectory of Remo-M. On the other hand, the path of the HWILS passed through way-points; however, the degree of following the path line was weaker than that of the Remo-M. Therefore, differences from the trajectory of the Remo-M occurred more than in the drone model results. The differences in trajectories can be reduced if finer tuning is performed. However, even with the above results, the use of HWILS for designing and verifying the drone redirection algorithm would be sufficient.

III. PROPOSED COMMERCIAL FIXED-WING DRONE REDIRECTION SYSTEM

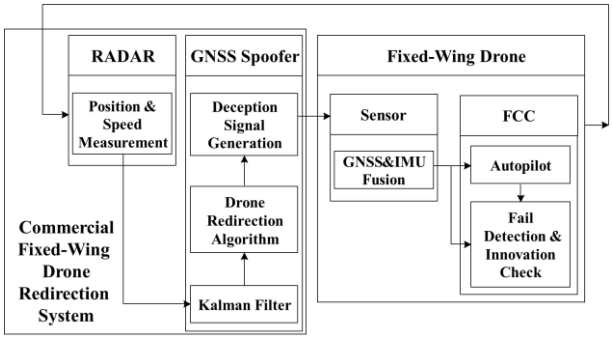

Fig. 5. Configuration of commercial fixed-wing drone redirection system.

We developed a drone redirection system, as shown in Fig. 5, using GNSS deception. The drone redirection system has a closed-loop structure that redirects the drone to a target position. The drone redirection system comprised a RADAR, which can detect the position and speed of the drone, and a GNSS spoofer, which generates GNSS deception signals to make the drone fly in a desired direction. In this study, the drone redirection system modeling was performed with the drone model detailed in Section II as the target to design and verify the drone redirection algorithm. The RADAR of the drone redirection system can be replaced with other sensors that can detect the position and speed of the drone; however, in this study, RADAR was selected because of its relatively long detection range.

The drone redirection system generates the GNSS deception position and velocity for drone redirection at every step based on RADAR measurement data in the drone redirection algorithm to change the direction of the drone and fly it to the target position.

A.RADAR

For synchronized GNSS deception, the position and speed of the drone were measured using RADAR, and GNSS deception was performed based on the measured position and speed from RADAR. Furthermore, continuous drone detection measurements from RADAR were required to redirect the drone to the target position.

As drone redirection system modeling is essential for the design of the redirection algorithm, error modeling of RADAR was also necessary. For detailed RADAR error modeling, RADAR cross-section and ground clutter modeling are performed according to the attitude of the drone. However, the RADAR we used in this study is the FIELDctrl Range model of APS, Poland [34], which does not provide signal-processing stage information, such as signal-to-noise ratio (SNR) and clutter. Therefore, simple RADAR modeling was performed based on the RADAR error measurements. For simple modeling, the RADAR error was assumed to have a Gaussian distribution. During flight, the position and speed of the Remo-M were measured using the RADAR, and the bias and variance of the RADAR measurements were calculated using the measured data.

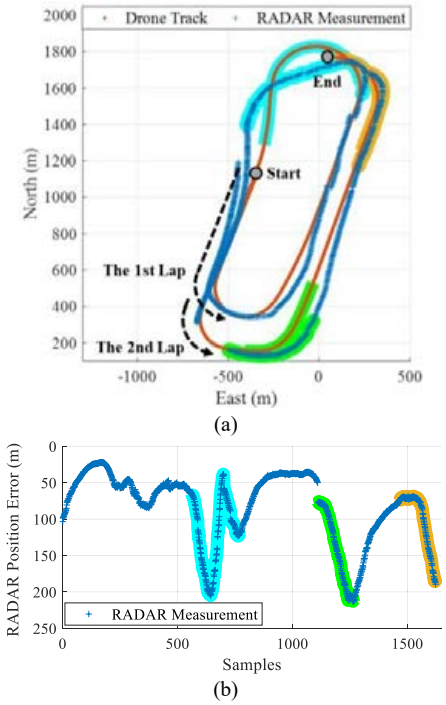

Fig. 6. RADAR position error according to flight trajectory. (a) Flight trajectory and RADAR measurement. (b) RADAR position error.

In the drone redirection scenario, the drone initially performed a straight flight and curved after GNSS deception was started. Fig. 6 shows the measurement results for RADAR error according to the flight trajectory. In Fig. 6(a), the blue cross marker indicates the RADAR measurement position, whereas the red line depicts the flight trajectory. Curved flight trajectories are color-coded. Fig. 6(b) shows that the error of RADAR increased when the drone flew in a curve rather than

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3264193

in a straight line. RADAR errors in curved flight are caused mainly by a mismatch between the maneuvering characteristics of the drone and the dynamic model of the Kalman filter used for RADAR tracking [35]. Therefore, the drone detection data of RADAR were acquired by switching to stick auto mode during Remo-M flight to perform a curved flight, to model the error during curved flight. At this time, the flight altitude of the drone was kept constant at 400 m AGL. RADAR error was calculated using the position and speed measurements from RADAR and GNSS logging data of Remo-M. For simple bias modeling, linear regression was performed on the measured RADAR error.

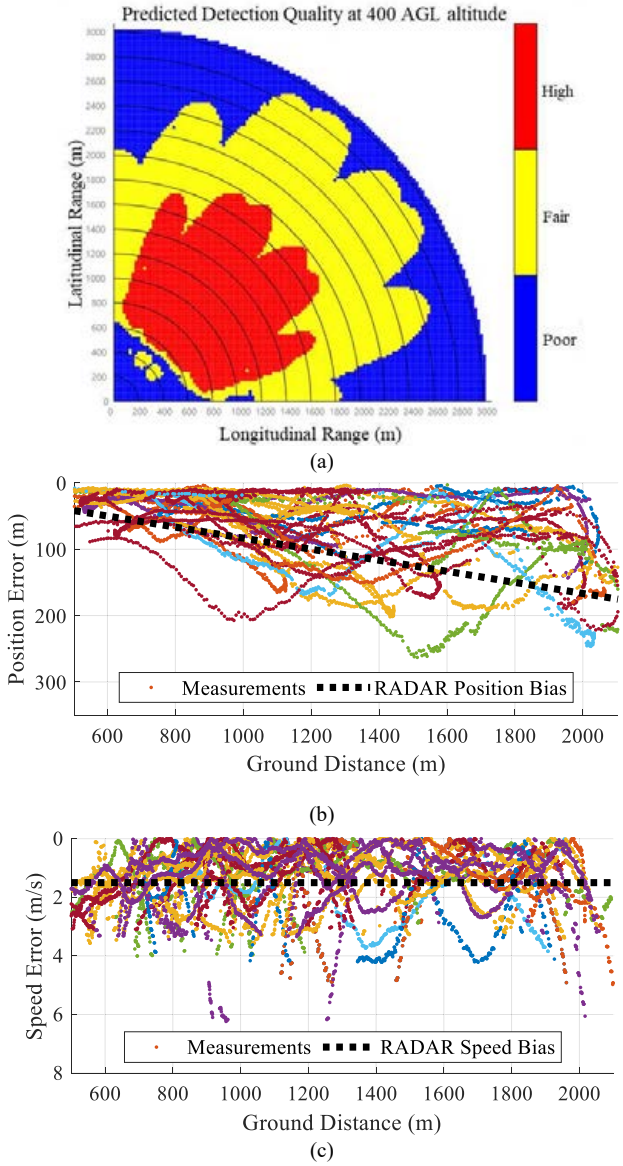

Fig. 7. RADAR position and speed errors according to ground distance. (a) Predicted Detection Quality at 400 m AGL. (b) RADAR position error. (c) RADAR speed error.

Fig. 7 shows the RADAR position errors and speed errors obtained through several flight tests. Fig. 7(a) shows the predicted detection quality (PDQ) obtained via APS's RADAR to check the ground clutter environment before RADAR operation. Based on the RADAR installation position, PDQ at a distance of up to 3000 m per axis is shown for 400 m AGL. Red, yellow, and blue regions indicate high, moderate, and poor PDQ, respectively. The flight test for RADAR error measurement was performed within the red and yellow areas of the PDQ. As shown in Fig. 7(b), the position bias is proportional to the ground distance, which is the horizontal plane distance between the RADAR installation and drone positions. To simulate a worse situation, the slope of the bias was set to 1/12, which is larger than the linear regression result of 1/16. As shown in Fig. 7(c), the speed bias was set to 1.5 m/s. Considering bias is modeled linearly, all remaining errors were included in the variance. However, when calculating the variance such that the residual bias is included, it becomes a significantly large value. Therefore, to partially remove the bias, the local variance was calculated at every certain interval based on the time axis, and the global variance was calculated using the average of the local variances. The position and speed standard deviations were set to 2 m and 0.15 m/s, respectively.

B. GNSS Spoofer

1) Deception Signal Generation

As shown in Fig. 8, the GNSS deception signal can be generated by receiving the authentic GNSS signal, changing the navigation message or applying a delay of the C/A code, and changing the Doppler shift.

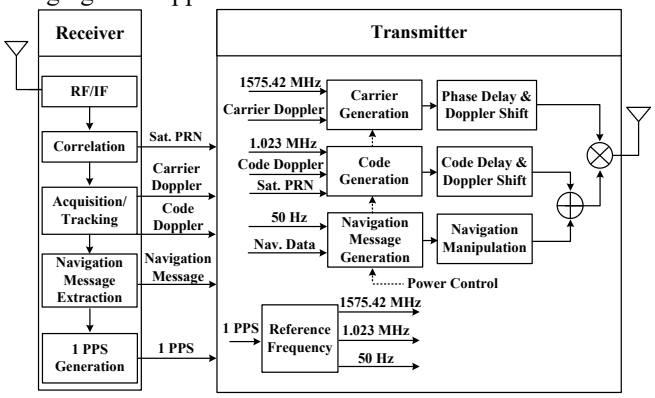

Fig. 8. Configuration for GPS L1 deception.

It is possible to change the navigation solution of the GNSS receiver by changing each time delay of the satellite signals by manipulating the handover word (HOW) of the navigation message, which is updated every 6 s [11]. This can cause a time delay of several tens of seconds, resulting in a position error of thousands of meters. However, when HOW is manipulated, the deception can be detected through the navigation message inspection using the internal clock of the receiver [36].

Alternatively, the time delay of the satellite signals can be manipulated using changing Doppler shifts and code delays, which can change the navigation solution [12]. In this study, the GNSS spoofer was designed to change the navigation solution in the above-mentioned manner. The GNSS spoofer was fabricated using a GNSS receiving antenna, navigation message extractor, time synchronization receiver, RF front end, FPGA and DSP, and GNSS transmitting antenna.

2) Kalman Filter

By using the EKF, the position and speed, which are output data of the RADAR, were received as input, and the output as position, velocity, and aperiodic measurement data of the RADAR were converted into periodic data.

RADAR measurement data are input at a rate of up to 20 Hz; however, this rate of input of the measurement data is not constant, and the data are input aperiodically. Additionally, rapid changes in the heading of the drone can result in tracking loss of the RADAR. If the Kalman filter is used, one can obtain the periodic data and use the estimation result of the Kalman

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3264193

filter in the case of RADAR tracking loss. As a result, periodic deception path calculation is made possible using the drone redirection algorithm.

3) Deception Position & Velocity Calculation for Redirection Concept

The concept of the drone redirection algorithm was established by considering the path-following algorithm of the fixed-wing drone. The drone generates lateral acceleration to minimize cross-track error. At this time, in the case of the fixed-wing drone, the lateral acceleration is determined based on the current heading measured by the drone's sensor and the desired heading. The current heading is affected by the velocity generated by the GNSS deception, whereas the desired heading is determined depending on the cross-track error and current position of the drone. The cross-track error is in turn affected by the position generated by the GNSS deception, that is, the $\vec{V}_k^{sensor}$ and $\vec{D}_k^{sensor}$ of (4) and (6) become the output of the GNSS receiver or the fusion result of GNSS and IMU. Therefore, $\vec{V}_k^{sensor}$ and $\vec{D}_k^{sensor}$ are similar to the deception velocity $\vec{V}_k^{sp}$ and $\vec{D}_k^{sp}$, where $\vec{D}_k^{sp}$ is the deception position vector, and ψ_k^{sp} is the direction of $\vec{V}_k^{sp}$.

The drone redirection concept can be divided into three cases, as shown in Fig. 9. For intuitive understanding, the deception velocity in Fig. 9 reflects the sample time.

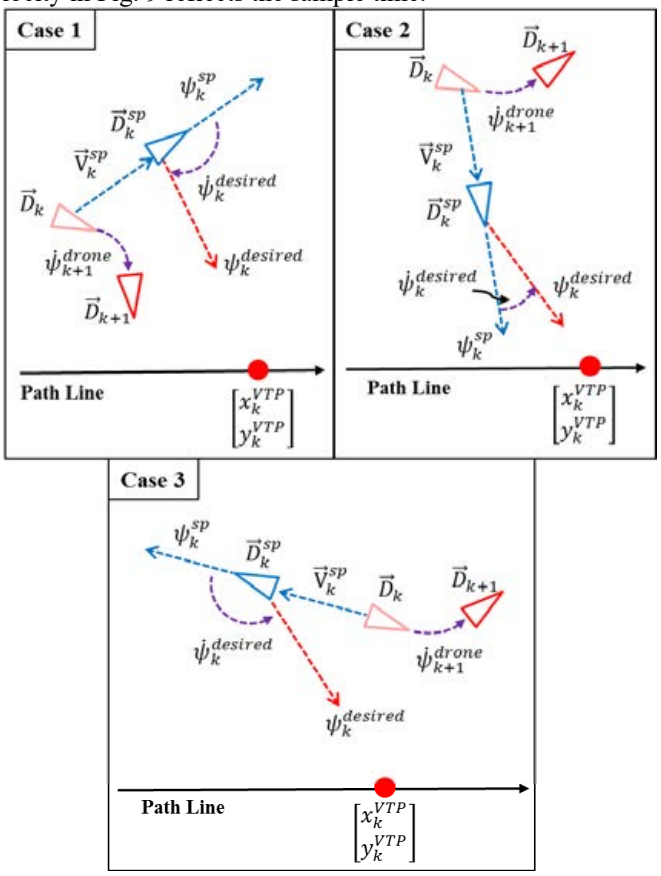

Fig. 9. Three cases for drone redirection concept.

$\psi_k^{desired}$ is determined by calculating the VTP by $\vec{D}_k^{sp}$. At this time, the drone recognizes ψ_k^{sp} as the drone's heading owing to $\vec{V}_k^{sp}$, and the difference between $\psi_k^{desired}$ and ψ_k^{sp} becomes $\dot{\psi}_k^{desired}$. In other words, it can be seen that $\vec{D}_k^{sp}$ and $\vec{V}_k^{sp}$ determine $\dot{\psi}_{k+1}^{drone}$. *Case 1* uses the deception position and velocity to make the drone fly in the clockwise direction, whereas *Cases 2* and *3* use the deception position and velocity to make the drone fly in the counter-clockwise direction. All cases can be applied symmetrically with respect to the path line. In case of symmetry, the direction of drone flight is applied in the opposite direction. The combination that can change the flight direction of the drone in the clockwise and counter-clockwise directions makes redirecting the drone possible. Therefore, *Case 1&2* or *Case 1&3* can be used for the drone redirection. Furthermore, it is possible to combine *Cases 1* and *1'* for the drone redirection. *Case 1'* means that *Case 1* is symmetrical with respect to the path line. In *Case 2*, after a certain time step, the deception position crosses the path line and operates the same as in *Case 1'*. Therefore, *Case 1&2* includes the combination *Case 1&1'*.

In this study, two strategies, *Cases 1&2* and *1&3*, were proposed for the drone redirection algorithm, and their analysis and verification were conducted.

4) Verification of Drone Redirection Concept

To verify the proposed redirection algorithm, we compared the simulation and experiment results from the flight test. When *Case 1* is verified, *Case 1'* is indirectly verified, and hence, *Case 2*, which includes *Case 1'*, is also partially verified. Therefore, only *Cases 1* and *3* were verified. In *Cases 1* and *3*, we checked whether the drone flew in the clockwise or counter-clockwise directions, respectively. Simulation was performed using Simulink of Mathworks with the configuration shown in Fig. 5. In addition, the simulation was performed without applying the RADAR error, and the flight test was conducted using the Remo-M, a commercial fixed-wing drone, with low power transmission in an area where civilians would not be harmed.

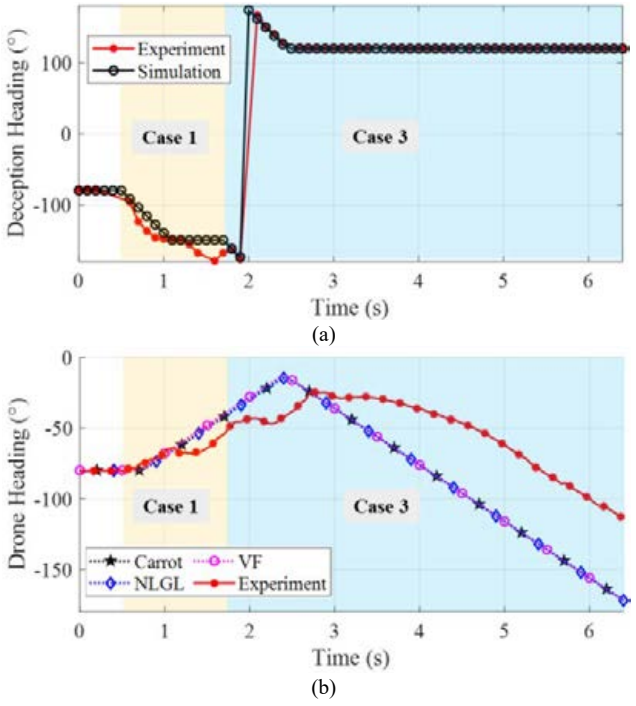

Fig. 10. Simulation and test results of the drone redirection concept. (a) Deception heading. (b) Drone heading.

Fig. 10 shows the simulation results and experiment results from the flight test. Fig. 10(a) shows the heading of the deception velocity, where the experiment result is obtained from the GNSS spoofer. Fig. 10(b) shows the heading of the drone, where the experiment result shows the yaw of Remo-M. During the simulation, the target drone model was simulated

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3264193

using a model that employs carrot chasing, NLGL, and VF (described in Section II). Remo-M in auto flight was initially flying in a −80 ° heading on the path line. The drone redirection algorithm operates without knowing the path line of Remo-M, and estimates the direction of the path line by observing the flight direction of Remo-M for a certain period of time. Then, a GNSS deception position and velocity are calculated based on the direction of the estimated path line. When the deception of *Case 1* began at 0.5 s, the flight direction of the Remo-M changed to the clockwise direction and deviated from the path line as the Remo-M's heading increased. The deception position and velocity changes from *Cases 1* to *3* from approximately 1.8 s, and *Case 3* is completely set at approximately 2.5 s. In the experiment result from the flight test, the flight direction changed in the counter-clockwise direction from approximately 2.8 s. Furthermore, it can be seen that the heading change slope gradually increased with time. This confirms, as a result of the simulation, that carrot chasing, NLGL, and VF have similar characteristics. The difference between the simulation and experiment results was caused by the omission of the inner loop controller of the drone model and delay of the response time of the GNSS receiver. Fig. 10 confirms that the tendency of the flight direction of Remo-M to change according to *Cases 1* and *3* was well represented.

C.Fail-detection & Innovation Check Analysis

The drone performs the fusion of IMU and GNSS using the Kalman filter. The difference between the estimated value based on the IMU navigation solution and the measured value based on GNSS, as shown in (13), is called innovation [37–40]. The drone monitors the magnitude of the innovation and performs a fail-detection and innovation check. If the innovation is larger than a specific value, the fusion of IMU and GNSS may not be performed. Such monitoring is referred to as an innovation check in this study. We analyzed fail-detection and innovation check on Ardupilot and PX4, which are widely used open-source software. Failsafe is performed to prevent accidents when one of the necessary functions seems to malfunction during drone flight. Therefore, fail-detection is performed to monitor the occurrence of various failures. Herein, the scope is limited to fail-detection for GNSS deception.

1) Fail-detection

Fail-detection is performed using specific thresholds and conditions for the innovation or the GNSS reception status. If a false occurs during fail-detection, a failsafe event is executed. In the case of the fixed-wing drone, the failsafe event sends a related failure message to the ground control station (GCS) in both Ardupilot and PX4; however, it does not change the flight mode in the default setting. Nonetheless, in the case of Ardupilot, there is a function related to failsafe events, which allows the user to add code to it, whereas in the case of PX4, there is a parameter that can be set to make the drone loiter at failsafe.

Ardupilot defines *posVar* and *velVar* and performs fail-detection accordingly. *posVar* and *velVar* are defined as

$$posVar = \sqrt{\frac{\mathrm{innovP_N^2} + \mathrm{innovP_E^2}}{\mathrm{P_{Pos_N}} + \mathrm{P_{Pos_E}} + Var_{\mathrm{Pos_N}} + Var_{\mathrm{Pos_E}}}}, \quad (18)$$

$$velVar = \sqrt{\frac{\mathrm{innovV_N^2} + \mathrm{innovV_E^2}}{\mathrm{P_{Vel_N}} + \mathrm{P_{Vel_E}} + Var_{\mathrm{Vel_N}} + Var_{\mathrm{vel_E}}}}, \quad (19)$$

The position and velocity components of innovation are *innovP* and *innovV*, respectively. In the north-east-down (NED) coordinate system, north and east are expressed as N and E, respectively, which are used in the subscripts of the variables. P is the same as that in (14), and *Var* is the measurement noise matrix of GNSS. The range of the *fail count* value is from 0–7; if the *fail count* is 7 or more, *bad variance* is defined as true, and the failsafe event function is executed, considering that the result of the fail-detection is false. If *bad variance* is true, it stays true until the *fail count* becomes 0. The detailed conditions of *fail count* are described in [18,27].

The fail-detection on PX4 checks the GNSS reception status, which refers to the standard deviation of the horizontal and vertical positions, number of satellites, standard deviation of the horizontal velocity, fix quality, etc. If the threshold value defined by the user is exceeded, a false is determined to have occurred during fail-detection.

In the case of HWILS, the user can add an event code when the result of the fail-detection is false; however, at default setting, no event is held except for sending a message to the GCS.

In the default setting, that is, in the case of the fixed-wing drone, Ardupilot, PX4, and HWILS trigger no events except that of delivering a false message to the GCS. However, it is possible for the users to add event code or to set parameters and for the GCS operator to change the flight mode; therefore, it is necessary to consider these possibilities in GNSS deception.

2) Innovation Check

In the case of Ardupilot, *posVar* and *velVar* are checked in an innovation check. If *posVar* is smaller than *GPSPosInnovGate*, the positions measured based on GNSS and estimated based on the IMU navigation solution are fused. Furthermore, *velVar* determines the velocity fusion by comparing the *GPSVelInnovGate* value. *GPSPosInnovGate* and *GPSVelInnovGate* are set to 5 by default and can be changed by the user.

In the case of PX4, unlike on Ardupilot, the maximum innovations of the north and east axes of innovation are compared as in (20) and (21). Therefore, PX4 has a slightly more sensitive innovation check compared to that on Ardupilot.

$$\mathrm{Max}\left(\frac{\mathrm{innovP_N^2}}{\mathrm{P_{Pos_N}} + Var_{\mathrm{Pos_N}}}, \frac{\mathrm{innovP_E^2}}{\mathrm{P_{Pos_E}} + Var_{\mathrm{Pos_E}}}\right) \leq innov\_gate^2 \quad (20)$$

$$\mathrm{Max}\left(\left(\frac{\mathrm{innovV_N^2}}{\mathrm{P_{Vel_N}} + Var_{\mathrm{Vel_N}}}, \frac{\mathrm{innovV_E^2}}{\mathrm{P_{Vel_E}} + Var_{\mathrm{Vel_E}}}\right) \leq innov\_gate^2 \quad (21)$$

When (20) is true, the positions measured based on GNSS and estimated based on the IMU navigation solution are fused. Equation (21) also determines whether to perform velocity fusion according to true or false. *Innov_gate* in (20) and (21) is also set to 5 as the default value, the same as on Ardupilot. This default value can be changed by the user. In other words, the result of the innovation check is false when the value of innovation exceeds 5 σ, and hence, GNSS and IMU are not fused. This is similar to the normalized innovation squared threshold of [15] and [16], which is set to 5 σ.

In the case of HWILS, the expression for innovation check is not disclosed; however, the operating principle is disclosed in the manual. HWILS checks the innovation of velocity and position using a gate window during the innovation check. If a false occurs during the innovation check, the check is performed by gradually increasing the gate window, and if a

true occurs, the gate window is gradually reduced and checked. Additionally, if the result of the innovation check is false for more than 5 s, GNSS and IMU fusion are forced to be performed, and innovation check is initialized.

If a false occurs in the innovation check, a delay occurs in the drone redirection via GNSS deception, as the drone performs dead reckoning.

3) Constraints for Deception Trajectory Generation

When generating a deception trajectory, constraints must be applied to the position, velocity, and acceleration of the deception trajectory to remain undetected by innovation check and fail-detection. Similar to [15,16], when calculating the next state from the current state of deception, constraints were applied to each position, velocity, and acceleration. In this study, the cases were divided according to whether the constraints were weak or strong.

4) Simulation Results

Using the two strategies proposed in this study, *Cases 1&2* and *1&3* of drone redirection to a target position were simulated, where the constraints were divided into weak and strong. At this point, fail-detection was performed using Ardupilot's algorithm and innovation check was performed using Ardupilot and PX4, respectively. Moreover, two cases in which the RADAR error was applied and not applied were simulated.

In the simulation, the variable values for Ardupilot were used. The acceleration standard deviation was 0.6 m/s^2, GNSS position standard deviation was 1 m, and the velocity standard deviation was 0.5 m/s. Similar to those applied on Ardupilot and PX4, the Kalman filter of the drone was set to operate at 400 Hz and GNSS at 10 Hz, and the drone redirection algorithm was set to operate at 10 Hz. GNSS deception was applied for 25 s from 7 to 32 s.

TABLE 1.
Simulation results of fail-detection and innovation check

	RADAR Error	Constraints	Fail-Detection (%)		Innovation Check (%)	
			Ardu	PX4	Ardu	PX4
CASE1&2	X	X	0.0	0.0	0.0	0.4
	X	Weak	0.0	0.0	0.0	0.0
	O	X	86.7	80.1	10.2	21.8
	O	Weak	52.3	57.6	1.7	7.4
	O	Strong	10.5	8.6	1.5	2.4
	Var. only	Strong	0.0	0.0	0.0	0.0
CASE1&3	X	X	10.7	9.1	3.2	4.0
	X	Weak	4.0	6.0	0.0	2.8
	X	Strong	4.8	4.8	0.0	0.0
	O	X	86.0	85.7	11.8	25.5
	O	Weak	86.2	87.3	1.3	6.9
	O	Strong	21.9	19.9	1.3	2.4
	Var. only	Strong	6.7	6.3	0.0	0.0

Table 1 shows the ratio of false durations for the fail-detection and innovation check to the duration of deception. In the calculation of the rate, an innovation check was considered if any one of the positions or velocities of the innovation was false. Considering the duration of false changes due to RADAR error, the simulation was repeated 100 times to obtain the average value. The innovation check confirmed that it is highly dependent on the RADAR error, and when the constraints increase, the false duration decreases. In addition, innovation check can be avoided by reducing the bias of the RADAR error, whereas fail-detection can be avoided in *Case 1&2*.

RADAR bias can be reduced by installing the RADAR with precise alignment, whereas tracking loss and bias error can be reduced when using multiple RADARs or fusion with other types of detection devices. Innovation also increases when the drone has a high flight speed. Moreover, in flight tests, delays occur for the GNSS spoofer, RADAR, and GNSS receiver mounted on the drone, which increase innovation. To reduce these, it is necessary to measure and compensate for their approximate delays.

D.Simulation Results of Drone Redirection System

For the system simulation of the drone redirection, way-point 1 (WP1) was set to the origin, whereas way-point 2 (WP2) was set to −3000 m on the north axis, and 0 m on the east axis. The target position was set between −3000 and 3000 m at a 500 m interval for both the north and east axes. RADAR error was applied during the simulation, and the distance was calculated as the minimum value of the L2-norm of the horizontal plane output position of the Kalman-filtered RADAR and the target horizontal plane position. The average distance for each target position obtained by repeating the simulation 10 times is shown in Fig. 11. The distance in Fig. 11 shows only the output below 100 m.

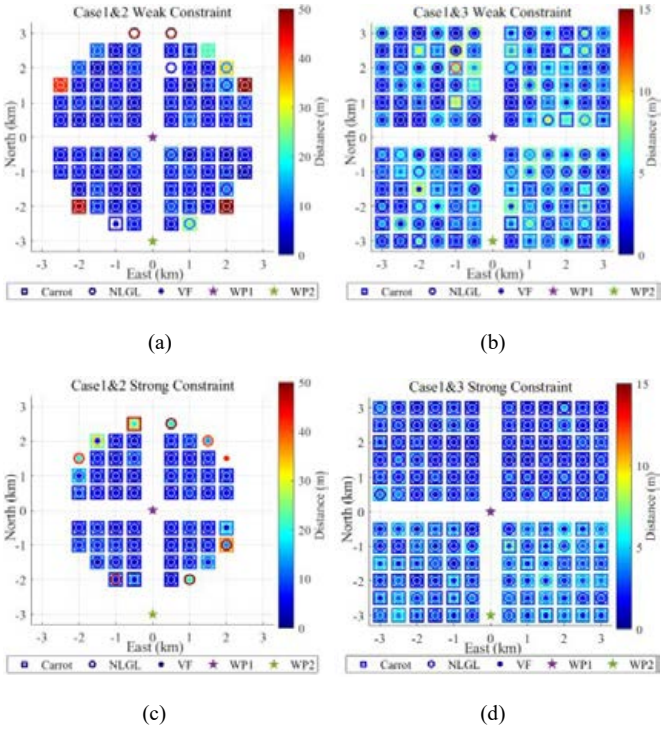

Fig. 11. Simulation results of drone redirection: range and accuracy. (a) *Case 1&2* weak constraints, (b) *Case 1&3* weak constraints, (c) *Case 1&2* strong constraints, and (d) *Case 1&3* strong constraints.

In *Case 1&2*, the innovation check and fail-detection did not result in significant detections; however, it is still possible to fly the drone to the target position set within the way-point radius. By contrast, in *Case 1&3*, the innovation check and fail-detection produced relatively significant detections; nonetheless, it is possible to fly the drone to a target position set

This article has been accepted for publication in IEEE Transactions on Aerospace and Electronic Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAES.2023.3264193

farther than the way-point radius.

Additionally, when generating a deceptive path, applying the strong constraints of *Case 1&2* narrows the range of possible target positioning. By contrast, for *Case 1&3*, the target positioning range does not narrow; however, upon the application of strong constraints, the success or failure of the drone redirection will depend on the algorithm of drones. As shown in Fig. 11, even if innovation check occurs, the drone is flown at a distance within 15 m of the target position. Most results of three path-following algorithms of the simple drone model are similar.

IV. EXPERIMENTS

A. Flight Test Results

The RADAR and flight test drone used the FIELDctrl Range from Poland APS and Remo-M, respectively. The flight test confirmed that the designed drone redirection system flew the automatically flying Remo-M to the target position. The test was carried out safely in a place where there could be no civilian damage. The goal distance refers to the horizontal plane distance between the target position and the Kalman-filtered RADAR measurement position in the NED coordinate system. The smaller the goal distance, the closer the drone to the target position.

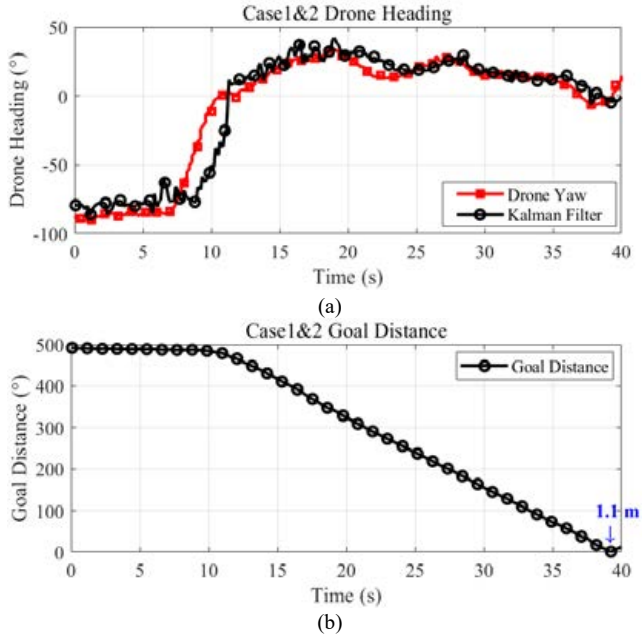

Fig. 12. *Case 1&2* test result of drone redirection. (a) Drone heading and (b) Goal distance.

Fig. 12 shows the result of the test performed by generating the GNSS deception path for *Case 1&2* with the drone redirection system. Fig. 12(a) shows the drone heading, where the red square and black circle graphs show the drone's yaw and the headings produced by the Kalman-filtered RADAR velocity measurement, respectively. Fig. 12(b) shows the goal distance. The drone initially flies in a heading of approximately −79 ° on the path line, and GNSS deception starts in 5 s, and the distance decreases to at least 1.1 m with time. It can be seen that the drone's flight direction is initially changed in the clockwise direction, and the drone's heading is gradually changed to fly toward the target position. Fig. 13 shows the results of the flight test using *Case 1&3*. The yellow area is the section where RADAR tracking is lost. As shown in Fig. 13(a), in *Case 1&3*, the direction change slope of the drone is larger than in *Case 1&2*, and hence, RADAR tracking is lost frequently. As shown in Fig. 13(b), because the RADAR tracking loss continued after approximately 78 s, the drone could no longer be flown close to the target position, and hence, the drone was kept loitering. The minimum goal distance of *Case 1&3* was measured at 145.6 m.

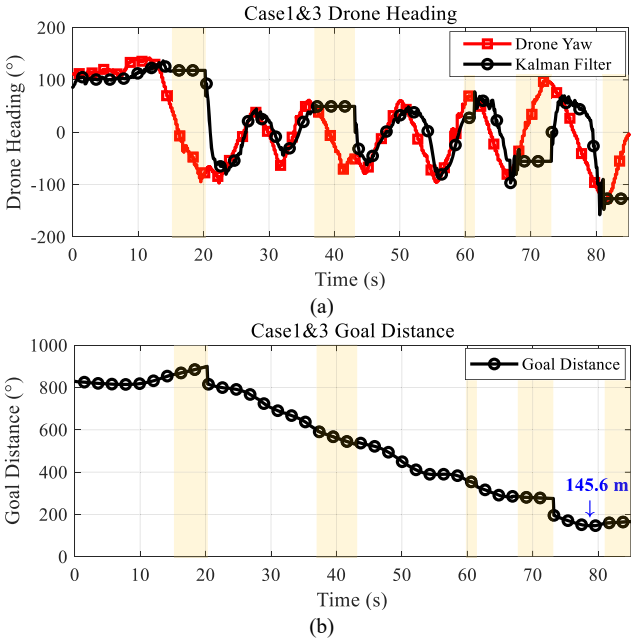

Fig. 13. *Case 1&3* test result of drone redirection. (a) Drone heading and (b) Goal distance.

B. Simulation Results using HWILS

To indirectly check whether the drone redirection algorithm is applicable to various drones, a simulation was performed using HWILS, which is Micropilot's trueHWIL^2 from Canada. As shown in Fig. 14, the deception position and velocity generated by the redirecting algorithm of the GNSS spoofer were output to the HWILS using UDP, and the position and velocity of the drone output from the HWILS were input to the Kalman filter of the GNSS spoofer through UDP.

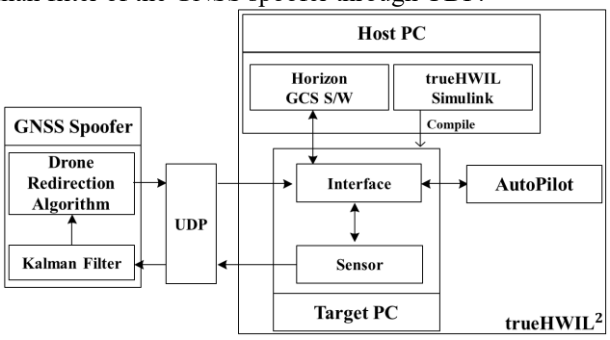

Fig. 14. Test configuration of the drone redirection algorithm in HWILS.

As per the results of the HWILS simulation of *Cases 1&2* and *1&3*, it was concluded that *Case 1&2* can redirect a drone and fly it to a target position, unlike *Case 1&3*. Therefore, the result of *Case 1&3* was conceptually analyzed, and a modified strategy for *Case 1&3* for HWILS was proposed.

1) Conceptual Analysis of Simulation Results

HWILS sets a virtual target heading (VTH) for the drone to direct the desired heading in only one direction without changing the direction between clockwise and counter-clockwise when the drone's heading differs by over

approximately 180° from the desired heading. Based on the desired heading, a semi-circle VTH area was determined to place the VTH, and the determined VTH area was not changed until the target way-point was reached. Therefore, when the HWILS simulation was performed using *Case 1&3*, the drone continued to rotate in one direction, as shown in Fig. 15. For intuitive understanding, the deception velocity in Fig. 15 reflects the sample time.

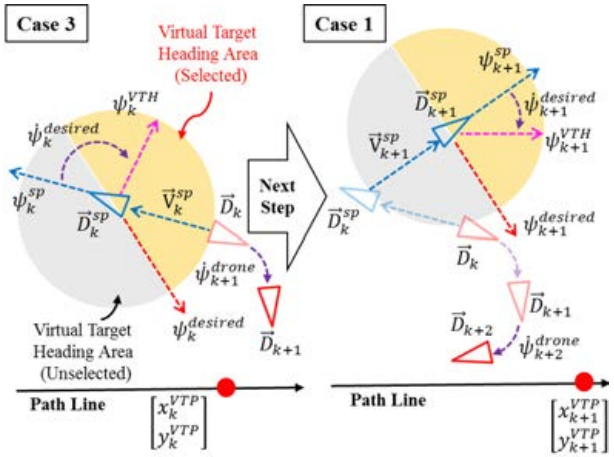

Fig. 15. Conceptual analysis of *Case 1&3* in HWILS.

2) Proposed Method for Fixed-Wing Drone Redirection

Case 2&3′ were modified and proposed to enable drone redirection in HWILS. As shown in Fig. 16, if the drone sets the VTH area using *Case 3*, and the GNSS spoofer generates a deception path using *Case 2&3′*, drone redirection is made possible.

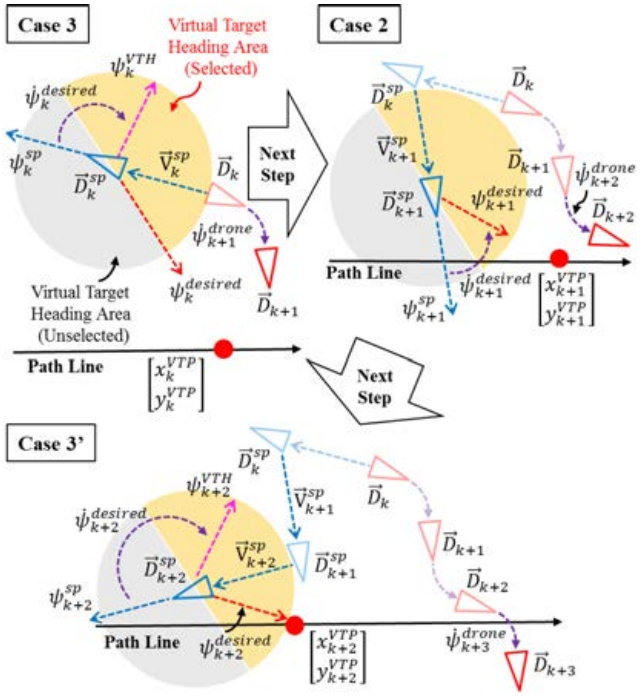

Fig. 16. Conceptual proposal of *Case 2&3'* in HWILS.

In this case, *Case 3′* indicates that *Case 3* is symmetrical with respect to the path line. For intuitive understanding, the deception velocity in Fig. 16 reflects the sample time. In the simulation using HWILS, RADAR error was not applied; the results are shown in Fig. 17. *Cases 1&2* and *2&3′* were both performed with weak constraints.

As shown in Fig. 17, from the results for the drone heading in (a) and goal distance in (c), the minimum goal distance of *Case 1&2* was found to be 0.4 m. At this point, no false occurred during the innovation check of the HWILS. On the other hand, the modified *Case 2&3′* was also a possible drone redirection method, as shown in the drone heading of (b) and goal distance of (d), and the minimum goal distance was found to be 1 m. At this point, there was an oscillation of the drone heading shown in Fig. 17(b), considering that some false errors occurred during the innovation check.

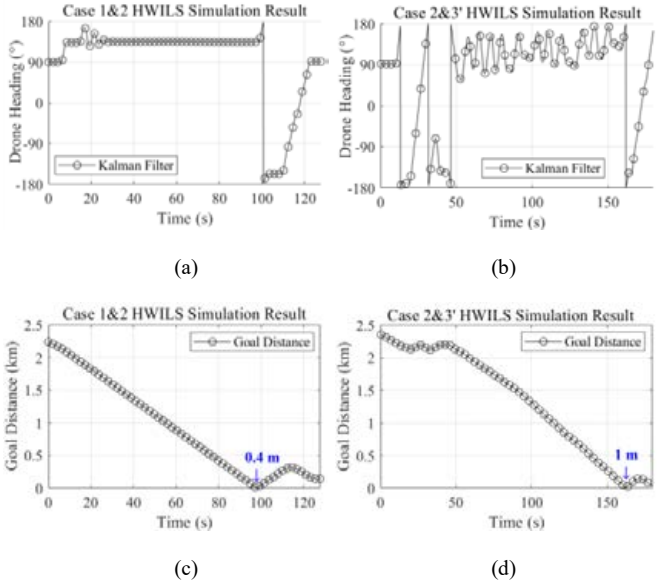

Fig. 17. Simulation results of drone redirection in HWILS. (a) *Case 1&2* drone heading, (b) *Case 2&3'* drone heading, (c) *Case 1&2* goal distance, and (d) *Case 2&3'* goal distance.

V. DISCUSSION

Owing to the increasing use of drones, anti-drone measures have become more crucial to preventing damage or abuse due to unlicensed drones. In this study, a drone redirection algorithm and system for preventing illegal intrusion of fixed-wing drones was proposed based on GNSS deception, one of the known anti-drone measures. Through simulations of simple drone modeling, flight tests, and simulations in HWILS, it was confirmed that the drone redirection algorithm and system are able to redirect a drone to a target position. Table 2 shows a comparison of the results of this study with those of related studies. As shown in Table 2, previous studies on drone redirection have been conducted for multi-rotors. By comparison, in this study, verification of the drone redirection algorithm was performed via simulation using simple drone modeling and HWILS, demonstrating in an outdoor test that redirection is possible, using a drone redirection system with a closed-loop structure equipped with RADAR. Therefore, the results of this study can fill the gap with regard to countermeasures against illegal intrusions by fixed-wing drones.

In addition, this paper proposed two strategies for redirection. In the case of *Case 1&2*, when compared to *Case 1&3*, the probability of drone fail-detection and innovation check detection is lower, and the probability of RADAR tracking loss is reduced through relatively slow drone flight direction change, but the redirection range depends on the position of the drone's way-point. At this point, in this study, the direction of the drone's path line can be estimated, but the way-point of the

TABLE 2. Comparison of results of this study with related studies

Related Study	Verification Method	Control Method	Target	Redirection Type	Prior Knowledge of Drone's Target Trajectory	Drone Detection Method
2019 [16]	Simulation	Closed-loop	Drone Model [1)]	To desired trajectory	Unknown	-
2019 [18]	Simulation	Closed-loop	Multi-rotor	Simple Redirection	Unknown	-
	Experiment	Open-loop			Initial Position	Human Eyes
2019 [19]	Simulation	Closed-loop		To target position	Unknown	-
	Experiment	HITL		Manually Control	Initial Hover Position	Human Eyes
2022 [20]	Experiment	Open-loop		Simple Redirection	Unknown	Human Eyes
2022 [21]	Simulation	Closed-loop		To target position	Known	-
2022 [22]	Simulation	HITL		Manually control	Unknown	-
	Experiment	Open-loop		Simple Redirection	Initial Hover Position	-
Ours	Simulation	Closed-loop	Fixed-Wing Drone	To target position	Unknown	RADAR
	Experiment					

1) Drone Model is not realistic. Target acceleration of the drone model is pre-planned regardless of the deception position and velocity.

drone is unknown; therefore, there is a limitation in that the redirection range cannot be unknown. By contrast, *Case 1&3* does not depend on the position of drone's way-point. However, *Case 1&3* may need modification depending on the algorithm of the drone. If the two strategies for the fixed-wing drone redirection presented herein are correctly used, it will be possible to automatically redirect a fixed-wing drone that invades a specific area or flies without permission. For example, if the distance between the core facility and the intruding drone is longer than a specific distance, *Case 1&2* can be used, whereas if the distance is within the specific distance, *Case 1&3* can be used. At this time, if the drone is expected to rotate in only one direction in *Case 1&3*, the strategy can be switched to *Case 2&3'*. Therefore, the results of this study can be used to effectively protect core facilities by controlling the flight directions of illegally intruding fixed-wing drones.

Considering that the performance of the drone redirection system was highly dependent on the RADAR performance, using a precisely aligned, installed RADAR combined with multiple RADARs or other types of detection devices helps reduce tracking loss and bias. This can be expected to improve the drone redirection performance, innovation check, and fail-detection avoidance performance.

In addition to redirection using GNSS deception, most methods for countering intruding drones require drone detection and identification to be performed first. Therefore, in the case of fixed-wing drones that cannot be detected and identified by RADAR, it is difficult to redirect the drones using GNSS deception. In the case of small drones that fly like birds, as in [41,42], it will be difficult to detect, track, and identify drones using drone detection RADAR, making it difficult to counter to illegal drone intrusion. To detect, identify, and track these drones, the sensor fusion of the different types of drone detection devices [43,44] must be studied. Consequently, each detection device should be designed accordingly.

If reinforcement learning is performed using the simplified drone model and HWILS presented in this study, it would be possible to develop a general-purpose and more optimized drone redirection algorithm. However, this needs to be further researched. Lastly, further research is needed for developing an algorithm that can redirect both fixed-wing and multi-rotor drones.

VI. CONCLUSION

In this study, a drone redirection system that redirects an illegally intrusive commercial fixed-wing drone in automatic flight to a target position using GNSS deception was developed. Using simple drone modeling that can easily change various characteristics, in conjunction with fail-detection and fusion conditions of drones based on a small number of parameters, the concept of the drone redirection algorithm was easily established in the initial stage of design. After tuning the simple drone model based on Remo-M flight test results, two strategies for redirection were established based on the simulation results for the simple drone model and flight test results on GNSS deception. The two strategies were implemented as a drone redirection algorithm, and a drone redirection system model was established by combining with a RADAR model based on RADAR measurement results obtained using FIELDctrl Range model of APS. Based on the simulation results for the drone redirection system model, it was confirmed that the simple drone model could be redirected to a target position located in any direction. Furthermore, the results of fail-detection and innovation check according to each strategy and constraints strength were demonstrated. In addition, a drone redirection system was constructed using a RADAR and GNSS spoofer, and it was demonstrated in a Remo-M flight test that drone redirection to a target position is possible. To indirectly confirm whether the drone redirection algorithm is applicable to various drones, a simulation was performed using Micropilot's HWILS. Then, based on the results of this study, the research findings, research improvement directions, limitations, and further research were discussed.

REFERENCES

[1] 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, 1 Jan. 2022, doi: 10.1109/MAES.2021.3115205

[2] 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

[3] D. Schmidt, K. Radke, S. Camtepe, E. Foo, and M. Ren, "A survey and analysis of the GNSS deception threat and

countermeasures," *ACM Comput. Surv.*, vol. 48, no. 4, pp. 1–31, May 2016

[4] Z. Renyu, S. C. Kiat, W. Kai and Z. Heng, "Deception attack of drone," *Proc. IEEE Int. Conf. Comput. Commun.*, pp. 1239–1246, Dec. 2018

[5] D. Shepard, J. Bhatti, and T. Humphreys, "Drone hack: Deception attack demonstration on a civilian unmanned aerial vehicle," *GPS World*, vol. 23, pp. 30–33, Aug. 2012

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

[7] L. Meng, L. Yang, W. Yang, and L. Zhang, "A Survey of GNSS spoofing and anti-spoofing technology," *Remote Sensing*, vol. 14, no. 19, p. 4826, Sep. 2022, doi: 10.3390/rs14194826

[8] N. Ji, Y. Rao, X. Wang, D. Zou, X. Chen, and Y. Guo, "Spoofing traction strategy based on the generation of traction code," *Remote Sensing*, vol. 15, no. 2, p. 500, Jan. 2023, doi: 10.3390/rs15020500

[9] Y. Gao and G. Li, "Two regional deployment algorithms of distributed GNSS forwarding spoofer for multiple receiver sensors," *Sensors*, vol. 22, no. 20, p. 7793, Oct. 2022, doi: 10.3390/s22207793

[10] N. Stenberg, E. Axell, J. Rantakokko, and G. Hendeby, "Results on GNSS spoofing mitigation using multiple receivers," *NAVIGATION: Journal of the Institute of Navigation,* vol. 69, no. 1, Jan. 2022. doi: 10.33012/navi.510

[11] Q. Yang, Y. Zhang, and C. Tang, "A novel GPS deception algorithm based on modifying navigation message," *Digit. Commun. Netw.*, vol. 236, 2017, doi: 10.1007/978-3-319-78130-3_6

[12] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, "Assessing the deception threat: Development of a portable GPS civilian spoofer," *Proc. Int. Tech. Meeting Satell. Div. Inst. Navig.*, pp. 2314–2325, 2008

[13] J. Bhatti and T. E. Humphreys, "Hostile control of ships via false GPS signals: demonstration and detection," *Navig. J. Inst. Navig*., vol. 64, no.1, pp. 51–66, Spring 2017, doi: 10.1002/NAVI.183

[14] D. Mendes, N. Ivaki, and H. Madeira, "Effects of GPS deception on unmanned aerial vehicles," *IEEE 23rd PRDC*, 2018, pp. 155–160, doi: 10.1109/PRDC.2018.00026

[15] A. J. Kerns, D. P. Shepard, J. A. Bhatti, and T. E. Humphreys, "Unmanned aircraft capture and control via GPS deception," *J. Field Robot.*, vol. 31, no. 4, pp. 617–636, 2014

[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 using low-cost SDR platforms," *Wireless Pers. Commun.*, vol. 115, pp. 2729–2754, 2020, doi: 10.1007/s11277-020-07211-7

[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 deception," *ACM Trans. Privacy Secur.*, vol. 22, no. 2, pp. 1–26, 2019, doi: 10.1145/3309735

[19] D. He, Y. Qiao, S. Chen, X. Du, W. Chen, S. Zhu, and M. Guizan, "A friendly and low-cost technique for capturing non-cooperative civilian unmanned aerial vehicles," *IEEE Netw.*, vol. 33, no. 2, pp. 146–151, Mar./Apr. 2019, doi: 10.1109/MNET.2018.1800065

[20] R. Ferreira, J. Gaspar, P. Sebastião, and N. Souto, "A software defined radio based anti-UAV mobile system with jamming and deception capabilities," *Sensors*, vol. 22, no. 4, 2022, doi: 10.3390/s22041487

[21] W. Chen, Y. Dong, and Z. Duan, "Accurately redirecting a malicious drone," *IEEE CCNC*, 2022, pp. 827–834, doi: 10.1109/CCNC49033.2022.9700664

[22] H. Sathaye, M. Strohmeier, V. Lenders, and A. Ranganathan, "An experimental study of GPS spoofing and takeover attacks on UAVs," in *31st USENIX Security Symposium*, 2022, pp. 3503–3520.

[23] Á .Gómez-Gutiérrez and G. R. Gonçalves, "Surveying coastal cliffs using two UAV platforms (multi-rotor and fixed- wing) and three different approaches for the estimation of volumetric changes," *Int. J. Rem. Sens.*, vol. 41, no. 21, pp. 1–3, Apr. 2020, doi: 10.1080/01431161.2020.1752950

[24] E. Ebeid, M. Skriver, K. Terkildsen, K. Jensen, and U. Schultz, "A survey of Open-Source UAV flight controllers and flight simulators," *Microprocess. Microsyst.*, vol. 61, pp. 11–20, 2018, doi: 10.1016/j.micpro.2018.05.002

[25] J. Wang, Y. Xiao, T. Li, and C. L. P. Chen, "Impacts of GPS deception on path planning of unmanned surface ships," *Electronics*, vol. 11, no. 5, 2022, doi: 10.3390/electronics11050801

[26] P. B. Sujit, S. Saripalli, and J. B. Sousa, "Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicles," *IEEE CSM*, vol. 34, no. 1, pp. 42–59, Feb. 2014, doi: 10.1109/MCS.2013.2287568

[27] Ardupilot Copter Project 2022 [online] Available: https://Ardupilot.org/plane/

[28] PX4 Open Source Autopilot [online] Available: https://px4.io/

[29] P. B. Sujit, S. Saripalli, and J. B. Sousa, "An evaluation of UAV path following algorithms," *ECC*, 2013, pp. 3332–3337, doi: 10.23919/ECC.2013.6669680

[30] S. Park, J. Deyst, and J. How, "A new nonlinear guidance logic for trajectory tracking," *Proc. AIAA Guid. Navig. Control Conf. Exhibit*, p. 4900, Aug. 2004, doi: 10.2514/6.2004-4900

[31] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for miniature air vehicles," *IEEE Trans. Robotics*, vol. 23, no. 3, pp. 519-529, Jun. 2007, doi: 10.1109/TRO.2007.898976

[32] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, "Vector field path following for small unmanned air vehicles," *2006 ACC*, Jul. 2006, p. 7, doi: 10.1109/ACC.2006.1657648

[33] MicroPilot trueHWIL [online] Available: https://www.micropilot.com/products-truehwilmp.htm/

[34] APS FIELDctrl Range 3D RADAR [online] Available: https://apsystems.tech/en/products/ultra-precise-3d-mimo-radars/

[35] W. Zhang, X. Zhao, Z. Liu, K. Liu, and B. Chen, "Converted state equation Kalman filter for nonlinear

maneuvering target tracking," *Signal Process.*, vol. 202, 2023, doi: 10.1016/j.sigpro.2022.108741

[36] R. Aanjhan, H. Ólafsdóttir, and S. Capkun. "SPREE: a deception resistant GPS receiver," *Proc. ICMCN-ACM*, New York, NY, USA, pp. 348–360, 2016, doi: 10.1145/2973750.2973753

[37] Ç. Tanıl, S. Khanafseh, M. Joerger, and B. Pervan, "An INS monitor to detect GNSS spoofers capable of tracking vehicle position," in *IEEE Trans Aerosp Electron Syst.*, vol. 54, no. 1, pp. 131–143, Feb. 2018, doi: 10.1109/TAES.2017.2739924

[38] M. L. Psiaki and T. E. Humphreys, "GNSS deception and detection," *Proc. IEEE*, vol. 104, no. 6, pp. 1258–1270, Jun. 2016

[39] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, "GPS vulnerability to deception threats and a review of antideception techniques," *Int. J. Navig. Obsev.,* vol. 2012, May 2012

[40] D. Mendes, N. Ivaki, and H. Madeira, "Effects of GPS deception on unmanned aerial vehicles," *Proc. IEEE 23rd Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, pp. 155–160, Dec. 2018

[41] W. He, X. Mu, L. Zhang, and Y. Zou, "Modeling and trajectory tracking control for flapping-wing micro aerial vehicles," *IEEE/CAA J. Autom. Sin.*, vol. 8, no. 1, pp. 148–156, Jan. 2021, doi: 10.1109/JAS.2020.1003417

[42] H. Huang, W. He, J. Wang, L. Zhang, and Q. Fu, "An all servo-driven bird-like flapping-wing aerial robot capable of autonomous flight," *IEEE/ASME Trans. on Mechatron.*, vol. 27, no. 6, pp. 5484–5494, Dec. 2022, doi: 10.1109/TMECH.2022.3182418

[43] J. Dudczyk, R. Czyba, and K. Skrzypczyk, "Multi-sensory data fusion in terms of UAV detection in 3D space," *Sensors*, vol. 22, no. 12, p. 4323, Jun. 2022, doi: 10.3390/s22124323

[44] K. Cisek, E. Brekke, M. Jahangir and T. A. Johansen, "Track-to-track data fusion for unmanned traffic management system," *2019 IEEE Aerospace Conference*, Big Sky, MT, USA, 2019, pp. 1-9, doi: 10.1109/AERO.2019.8741727

Myoung-Ho Chae received B.S. and M.S. degrees from ChungNam National University, Korea, in 2012 and 2014, respectively. He is currently a Ph.D. student at the Korea Advanced Institute of Science and Technology, Daejeon, Korea. He is currently a senior researcher at the Agency for Defense Development, Daejeon, Korea. His research interests include wideband frequency synthesizer, wideband receiver, and electronic warfare systems.

Seong-Ook Park (M′05) was born in KyungPook, Korea, in December 1964. He received a B.S. degree from KyungPook National University, Korea, in 1987, an M.S. degree from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1989, and a Ph.D. degree from Arizona State University, Tempe, in 1997, all in electrical engineering. From March 1989 to August 1993, he was a Research Engineer with Korea Telecom, Daejeon, working with microwave systems and networks. He later joined the Telecommunication Research Center, Arizona State University, until September 1997. Since October 1997, he has been with the Information and Communications University, Daejeon, and is currently a Professor at the Korea Advanced Institute of Science and Technology. His research interests include mobile handset antenna and analytical and numerical techniques in electromagnetics. Dr. Park is a member of Phi Kappa Phi.

Seung-Ho Choi received a B.S. degree from YeungNam University, S. Korea, in 1992, an M.S. degree from Pohang University of Science and Technology in 1998, and a Ph.D. degree from the Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 2008 all in electrical engineering. He later joined the Agency for Defense Development, Daejeon, Korea and is currently a Principal Researcher in ADD. His research interests include electronic attack technologies in the EW area.

Chae-Taek, CHOI received his B.S. and M.S. degrees in computer science and statistics from ChungNam National University, Korea, in 1989 and 1991, respectively. He is currently a principal researcher at the Agency for Defense Development since 1991. His research interests include neural network optimization, RF-Counter Unmanned Aerial system technology, and electronic-warfare system.