






# Hijacking of unmanned surface vehicles: A demonstration of attacks and countermeasures in the field

Petter Solnør<sup>1</sup>  | Øystein Volden<sup>1</sup>  | Kristoffer Gryte<sup>1</sup>  |  
Slobodan Petrovic<sup>2</sup>  | Thor I. Fossen<sup>1</sup> 

<sup>1</sup>Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway

<sup>2</sup>Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Gjøvik, Norway

## Correspondence

Petter Solnør, Department of Engineering Cybernetics, Norwegian University of Science and Technology, 7491 Trondheim, Norway.  
Email: [petter.solnor@ntnu.no](mailto:petter.solnor@ntnu.no)

## Funding information

Research Council of Norway,  
Grant/Award Number: 223254

## Abstract

Driven by advances in information and communication technologies, an increasing number of industries embrace unmanned and autonomous vehicles for services, such as public transportation, shipping, mapping, and remote surveillance. Unfortunately, these vehicles are vulnerable to passive and active cyber-physical attacks that can be used for industrial espionage and hijacking attempts. Since attackers can use hijacked vehicles as weapons in terrorist attacks, ensuring the secure operation of such vehicles is critical to prevent the attacks from causing dire financial consequences, or worse, the loss of human lives. This study is motivated by the observation that most cybersecurity studies provide superficial, high-level descriptions of vulnerabilities and attacks, and the true impact of the described attacks remains unclear. To address this problem, we demonstrate advanced manipulation attacks against an underactuated Unmanned Surface Vehicle (USV) which results in successful hijackings. Using state-of-the-art cryptography, we also show how the signal transmission can be secured to avoid hijacking attempts actively steering the vehicle off course. Through field experiments, we demonstrate how the attacks affect the closed-loop guidance, navigation, and control system and how the proposed countermeasures prevent these attacks from being successful. Our study is unique in that we provide a complete description of the attacked USV and give a detailed analysis of how spoofed navigation estimates affect the closed-loop behavior of the underactuated USV.

## KEYWORDS

authentication, cryptography, cyber-physical attacks, cybersecurity, encryption, marine robotics, unmanned surface vehicles

## 1 | INTRODUCTION

With an increased focus on autonomy, the autonomous ships market has become a multibillion dollar industry and is expected to face significant growth in the coming years (Jadhav & Mutreja, 2020).

Leveraging advanced information and communication technologies (ICTs), unmanned and autonomous vehicles have shown significant advantages for services, such as public transportation, environmental monitoring, mapping, and remote surveillance and are predicted to play an essential role in the future (Felski & Zwolak, 2020). Industrial

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2022 The Authors. *Journal of Field Robotics* published by Wiley Periodicals LLC.

leaders are racing to develop advanced autonomous solutions for ferries (Rolls-Royce, 2018) and cargo ships (Quinton, 2021), respectively. Additionally, commercialization of ideas from public research projects, for example, the Autoferry project (NTNU, 2021), occurs through spin-off companies seeking to develop autonomous ferries for urban public transportation.

Unfortunately, cybersecurity concerns threaten the growth of the autonomous ships market (Research and Markets, 2021). Hijacking attacks of autonomous ships pose a crucial threat, as they may be used for stealing goods or as weapons in terrorist attacks. Targeting other vessels or off-shore and coastal installations, for example, cruise ships, oil & gas platforms, and on-shore centers, such attacks threaten the lives of civilians and may cause dire financial consequences (Vinnem & Utne, 2018). Consequently, several challenges remain before fully autonomous ships can be accepted by authorities, classification societies, and the general public.

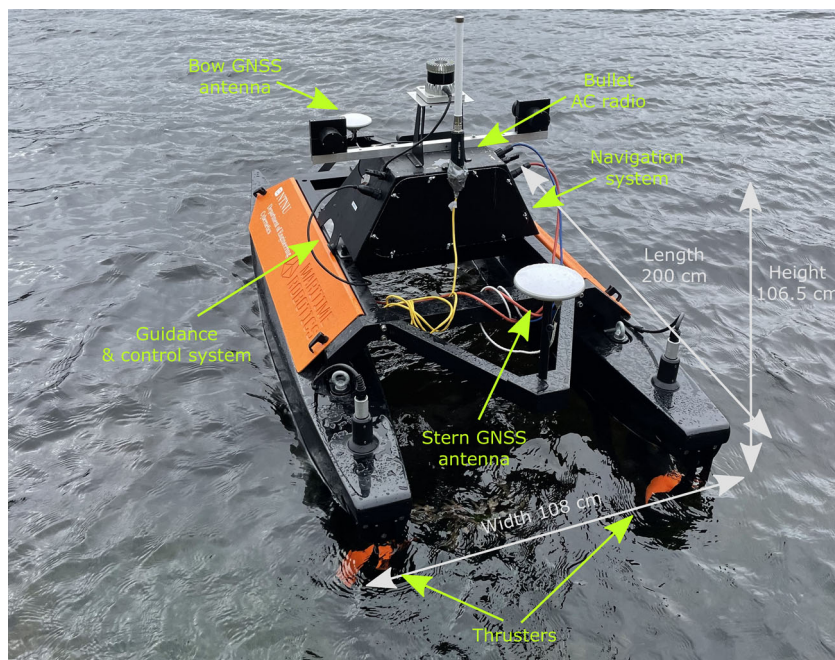
At the core of autonomous vehicles are advanced guidance, navigation, and control (GNC) systems (Fossen, 2021). Often implemented as distributed systems, the GNC components communicate over buses and networks spanning the vehicle. Historically, Controller Area Network (CAN) buses have been used for this purpose; however, Ethernet is becoming an increasingly popular option for intravehicular communication (Tuohy et al., 2015; Wollschlaeger et al., 2017). Generally, we refer to feedback control systems closing the loop over networks as Networked Control Systems (NCSs) (Zhang et al., 2020). With the ease of installation and reduced maintenance costs due to flexible software and hardware architectures, NCSs provide significant advantages over systems with independent communication channels (Hespanha et al., 2007). Nevertheless, these communication lines are inherently insecure, making NCSs vulnerable to cyber-physical attacks. Additionally, developers often use middleware frameworks such as the Robot Operating System (ROS) and

the Underwater Systems and Technology Laboratory (LSTS) toolchain (Pinto et al., 2013) to implement NCSs. In fact, according to a study by ABI Research (2019), ROS is expected to be present in a large fraction of future commercial robotic systems. However, these frameworks do not provide additional security mechanisms, and researchers have expressed concerns about the security of these frameworks for some time (Dieber et al., 2020; Teixeira et al., 2020). Therefore, it is essential to address these vulnerabilities, and as such, ROS 2, currently under development, includes additional security mechanisms (Fazzari, 2021).

While researchers have expressed concerns over the security of intravehicular communication, attacks taking advantage of the vulnerabilities are rarely demonstrated. This may lead to a false sense of security among system developers when using popular software frameworks. As a result, in this paper, we describe and demonstrate how we can exploit these vulnerabilities to hijack and take control of an underactuated unmanned surface vehicle (USV), thus bridging the gap between theory and practice. We also demonstrate how we can prevent these attacks by securing the GNC communication with modern cryptographic algorithms. The experiments are performed on the NTNU Otter USV shown in Figure 1.

## 1.1 | Related work

Because of the great benefits associated with NCSs, they are increasingly used in vehicles (El-Rewini et al., 2020). However, since NCSs connect system components across a network and are vulnerable to cyber-physical attacks, such as eavesdropping and data injection (Teixeira et al., 2012; Wang & Yang, 2019), researchers have expressed concerns about the cybersecurity of NCSs for many years (Dzung et al., 2005). In particular, with increased self-governance,



**FIGURE 1** An overview of the NTNU Otter unmanned surface vehicle [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

security breaches in onboard communication systems may directly cause altered behavior in unmanned and autonomous vehicles. As such, there is a growing concern about the cyber-physical resilience of these vehicles (Bolbot et al., 2020; Silverajan et al., 2018; Tan et al., 2020), and it is therefore critical to establish secure communication between the connected devices.

Numerous surveys and review papers have described vulnerabilities and cyber-attacks against vehicles. El-Rewini et al. (2020) describe vehicular cybersecurity challenges using a hierarchical framework to isolate threats and attacks in three layers; sensing, communication, and control. Considering a broad scope of attack vectors against inter- and intravehicular communication, Sun et al. (2021) discuss cybersecurity vulnerabilities related to autonomous cars. Similarly, in the maritime domain, Silverajan et al. (2018) describe relevant attack surfaces for unmanned smart ships. These attack surfaces, and cyber-attacks against autonomous ships, were later analyzed and classified according to the Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service (DoS), and Elevation of privilege (STRIDE) approach by Kavallieratos et al. (2019). With a focus on intravehicular communication, Yağdereli et al. (2015) describe attacks targeting communication lines, such as passive eavesdropping and active masquerading and message modification attacks, against unmanned and autonomous vehicles. Notably, these studies provide superficial descriptions, and the viability of executing the attacks, and the resulting consequences, remain unclear.

Considering cyber-attack demonstrations on intravehicular communication, Kang et al. (2018) implemented an attack against a CAN bus in a conventional car, where messages were first eavesdropped upon and analyzed, followed by the injection of spoofed messages. In the maritime domain, Lund et al. (2018) demonstrated an attack on an integrated Inertial Navigation System (INS) solution, where the estimated position of the vessel was changed by spoofing National Marine Electronics Association (NMEA) messages coming from the Global Navigation Satellite System (GNSS) receiver. While the CAN bus attack was implemented in a controlled laboratory setup, the INS attack was performed in the field with results visible on an Electronic Chart Display and Information System. Nevertheless, conventional, manned vehicles were the target of both attacks. Hence, the signals are not used directly in closed-loop control. As such, we find that the literature lacks studies demonstrating how unmanned and autonomous vehicles with increased self-governance are affected by such attacks.

To detect cyber-attacks against intravehicular communication, we can use cryptographic methods or anomaly-based intrusion detection systems (IDSs). The use of anomaly-based IDSs is often motivated by claims stating that encryption and authentication methods conflict with the link-layer data frames used or are too resource-intensive (Han et al., 2021; Wu et al., 2020). However, these assumptions may be problematic in many practical applications. First, anomaly detection methods are problematic themselves because they require accurate definitions of normality. This is very challenging, causing anomaly detection methods to suffer from high false-positive rates (Jallad et al., 2020). A high false-positive rate, combined

with a low probability of attack, that is, *base rate*, is problematic because of the base rate fallacy phenomenon (Axelsson, 2000). For this reason, anomaly-based IDSs are rarely used in practice (Jallad et al., 2020). Second, regarding the use of cryptographic algorithms, we argue that the cryptographic algorithms rarely have to be used at the link layer. Just like cryptographic operations are not applied on the payload of Ethernet frames, they need not be applied on the payload of CAN bus frames. Instead, they can often be used higher up in the communication protocol stack, for example, at the application layer. Concerning the efficiency of cryptographic algorithms, we find that modern, symmetric cryptographic algorithms are very efficient and can, therefore, be applied to feedback control systems without inducing significant time delays (Volden et al., 2021). For example, Mun et al. (2020) have suggested using cryptographic authentication methods on a CAN bus and conducted laboratory experiments for validation that demonstrated their efficiency.

## 1.2 | Main contributions

Rather than reiterating high-level descriptions of cyber-physical attacks and related countermeasures, the main objective of this study is to demonstrate that cyber-physical attacks can indeed be implemented and used to hijack a USV. We also show that our proposed cryptographic methods can prevent these attacks from being successful. In particular, we describe how manipulation of yaw (i.e., heading) and position estimates changes the behavior of an under-actuated USV. We proceed by describing how these attacks can be implemented and then suggest countermeasures that secure the GNC communication against eavesdropping, injection, and replay attacks. Finally, we implement the attacks on the insecure and the secured system and conduct field experiments to verify that the attacks are indeed successful in hijacking the vehicle without cryptographic protection and that the cryptographic methods successfully detect and prevent such attacks. The proposed cryptographic methods are beneficial compared with previously proposed anomaly-based IDSs because the problems of high false-positive rates and false-negative rates are reduced to a minimum if symmetric cryptographic algorithms are used. Consequently, contrary to anomaly-based IDSs, the proposed methods are appropriate for practical applications. In summary, the following are considered the main contributions of this study:

- We describe and analyze how manipulation of position and heading estimates affect the closed-loop behavior of an under-actuated USV.
- We provide a detailed description of how these attacks can be implemented.
- We describe how cryptographic methods can prevent such attacks and argue that they are more practical than previously proposed anomaly-based IDSs.
- We implement and demonstrate the effect of the described attacks and defensive measures on a USV.

### 1.3 | Outline

The remainder of this paper is structured as follows. In Section 2, we introduce cryptographic concepts relevant to securing distributed GNC systems. We then present the case study in Section 3, where we introduce USV motion control. On the basis of this, we describe how eavesdropping and spoofing attacks can be used to manipulate the USV and how cryptographic measures can prevent these attacks. In Section 4, we show the experimental setup and describe the experiments. Then, in Section 5, we describe and discuss the experimental results. Finally, Section 6 concludes the paper.

## 2 | CRYPTOGRAPHY

When a USV uses a distributed GNC system, it becomes vulnerable to cyber-attacks if adversaries gain access to the transmission lines. In fact, the usual assumption in security analysis is that adversaries do have access to the transmission lines. For example, an adversary with such access can eavesdrop on the communication to obtain confidential information or inject spoofed messages to manipulate the behavior of the USV. Such attacks may be used for industrial espionage and hijacking purposes. By using cryptographic methods, we can prevent these attacks from being successful.

### 2.1 | Cryptographic concepts and terminology

Cryptography is typically used to achieve secure signal transmission (confidentiality) across insecure communication lines. Today, in the analysis and design of cryptographic algorithms, it is assumed that the cryptographic algorithm is known by the adversary, and only the keys, and material directly derived from the keys, are kept secret. This is commonly referred to as *Kerckhoff's Principle*.

#### 2.1.1 | Symmetric and asymmetric cryptography

Cryptographic schemes are classified as *symmetric* and *asymmetric*, depending on whether the transmitter and the receiver use the same keys or not. Asymmetric cryptographic schemes are often based on number-theoretic problems that are believed to be hard, such as finding the prime factorization  $p, q \in \mathbb{N}$  of a composite number  $N = p \cdot q \in \mathbb{N}$ , where  $p$  and  $q$  are of approximately the same size (in bits), or finding the discrete logarithm  $b$  of a group element  $a = g^b \in \mathbb{G}$  given the very large group  $\mathbb{G}$ , the group element  $a$ , and the generator of the group  $g$ . On the other hand, symmetric cryptographic schemes are built using finite state automata, bitwise operators, such as *AND*, *OR*, and *XOR*, and transpositions and highly nonlinear substitutions. Consequently, symmetric cryptography is much faster than asymmetric cryptography in software. However, asymmetric cryptography brings other unique properties, such as the possibility of nonrepudiation and symmetric key exchange. Since the GNC components are assumed to be trusted entities and key

exchange is not required, these properties are unnecessary. As such, we will only consider symmetric cryptography in this paper.

#### 2.1.2 | Encryption

Encryption algorithms are used to obtain confidential signal transmission over insecure transmission channels. We refer to an encryption algorithm as a *block cipher* or a *stream cipher* depending on whether the algorithm is *stateless* or *stateful*. While block ciphers are  $N$ -bit substitutions parameterized by a secret  $K$ -bit key, the stream ciphers work by extending the key to a much longer pseudorandom sequence known as the *keystream*. Since the encryption algorithms need to work across insecure transmission channels, the stateful stream ciphers require a cryptographic synchronization mechanism. This is typically achieved using a public parameter known as the *initialization vector* (IV). The IV and the secret key are used to derive an initial state of the cipher, typically on a per-message basis. The input to an encryption algorithm is called *plaintext*, while the resulting output is called *ciphertext*. By decrypting the ciphertext, the corresponding plaintext is recovered. Without access to the secret key, the ciphertext should be computationally indistinguishable from white noise. An encryption algorithm is considered broken if an attack that recovers the key and/or the plaintext with computational complexity less than  $2^K$  exists. Today, a key size of 128 bits or more is recommended for data that needs to be protected after 2030 (Barker & Roginsky, 2019).

#### 2.1.3 | Authentication

Unfortunately, encryption does not ensure the integrity nor confirmation of the true origin of the message, that is, asserting that information received is from a trusted source. This is referred to as *data origin authenticity*. Data origin authenticity may be obtained through the use of *message authentication codes* (MACs). A MAC is a function parameterized by a secret, shared key that maps a message of arbitrary size to a fixed  $B$ -bit output. The output of the MAC is referred to as a *tag* and is transmitted with the message. Upon reception, the receiver, in possession of the secret key, recomputes the tag and compares the tag with the received tag. If the tags match, the message is considered authentic. In addition to resistance against key recovery attacks, a MAC should resist *existential forgery attacks*, that is, it should be infeasible for an adversary without knowledge of the secret key to produce a valid (message, tag)-pair for a new message. Assuming the MAC used is cryptographically secure, the computational complexity of an existential forgery is  $2^{\frac{B}{2}}$  because of the *birthday attack* (Stinson & Paterson, 2018, p. 143). Consequently, a tag size of 128 bits results in 64-bit security against existential forgery. The key size used in the MAC should be similar to that used in encryption algorithms, while the tag size depends on other considerations, such as the feasibility of testing large quantities of (message, tag)-pairs for the adversary. The most commonly used MAC is the Keyed-Hash Message Authentication Code, which constructs a MAC from cryptographic hash functions (Dang, 2008).

### 2.1.4 | Authenticated encryption

Since both confidentiality and data origin authenticity are desirable properties, encryption and MACs are often combined. This is referred to as *authenticated encryption*. Authenticated encryption can be obtained through the use of *generic compositions* such as “encrypt-then-MAC” (Bellare & Namprempre, 2008) or through dedicated algorithms designed to provide both confidentiality and data origin authenticity directly, such as AEGIS (Wu & Preneel, 2014).

## 2.2 | Fault checks and cryptographic authenticity

Before continuing, we emphasize the difference between conventional *fault checks* and cryptographic MACs. Fault checks such as *parity bits*, *checksums*, *cyclic redundancy checks* (CRCs), and *hash codes* are public, *unkeyed* algorithms designed to detect *inadvertent transmission errors* or *data integrity breaches*. As such, anyone with knowledge of the specific fault check used can forge valid messages. This is fundamentally different from MACs, for which it should be computationally infeasible for an adversary to compute a valid (message, tag)-pair for a new message, that is, an existential forgery.

Communication protocols frequently use conventional fault checks to discard corrupted messages. However, the existence of such fault checks does not make the system secure against active adversaries. These adversaries can forge valid messages that the receiver accepts. Examples of frameworks that use conventional fault checks and not cryptographic MACs include the InterModule Communication (IMC) protocol, used in the LSTS toolchain. Other frameworks, such as ROS, do not even use conventional fault checks (Dieber et al., 2020).

## 3 | CASE-STUDY: ATTACKING AND SECURING A USV

We proceed by introducing motion control systems for underactuated USVs. On the basis of this, we show how we can spoof the heading and the position to cause predictable changes in the

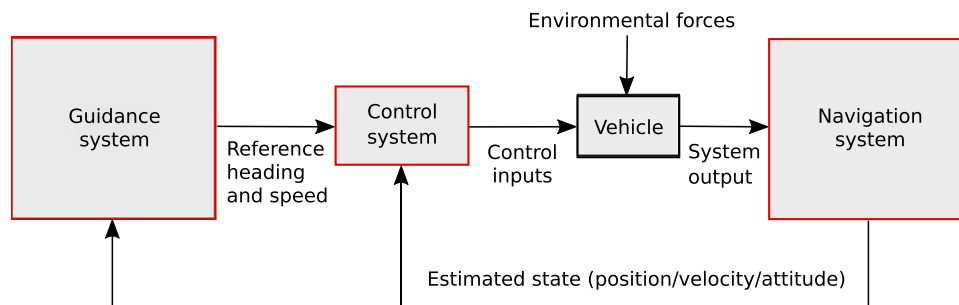
paths of USVs, illustrating that both are means of hijacking. We proceed by describing the technical implementation of the spoofing attacks. Finally, we show how cryptographic methods can be used as countermeasures to prevent such attacks.

### 3.1 | USV motion control

Let  $\eta = [N, E, \psi]^T \in \mathbb{R}^2 \times \mathbb{S}$  describe the vehicle pose and  $v = [u, v, r]^T \in \mathbb{R}^3$  describe the vehicle velocity in the earth-fixed North-East-Down (NED) reference frame and the body-fixed frame, respectively. To control the USV, a motion control system consisting of three independent system blocks, *guidance*, *navigation*, and *control*, is usually used. Notably, many USVs have two controls, for example, a propeller and a rudder. Consequently, these USVs can only directly control  $u$  and  $\psi$ , that is, surge speed and yaw, and are, therefore, underactuated. The navigation system estimates the position, velocity, and attitude of the USV, for example, by using GNSS receivers and an Inertial Measurement Unit (IMU), and the guidance system uses these estimates and the desired path to compute the desired yaw and the desired surge speed of the USV. The control system then uses the estimates from the navigation system and the desired yaw and surge speed from the guidance system to allocate thrust to the actuators of the USV. The signal flow between the GNC components is shown in Figure 2.

### 3.2 | Vehicle manipulation

Underactuated USVs usually solve the path following problem by defining a two-dimensional (2D) workspace consisting of along-track and cross-track errors and then using a line-of-sight (LOS) guidance law to minimize the cross-track error (Fossen, 2011, p. 258). Let the variables  $\psi$ ,  $\hat{\psi}$ , and  $\psi_d$  denote the true yaw, the estimated yaw, and the desired yaw of the vehicle, respectively. The true position of the USV is denoted by  $p^n = [x, y]^T$ , the estimated position of the USV is denoted by  $\hat{p}^n = [\hat{x}, \hat{y}]^T$ , and we assume that the USV is following a straight-line path, implicitly defined by the two waypoints (WPs)  $p_k^n = [x_k, y_k]^T$  and



**FIGURE 2** A generic motion control system for an underactuated unmanned surface vehicle [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



$p_{k+1}^n = [x_{k+1}, y_{k+1}]^T$ . Moreover, we consider a path-fixed reference frame, rotated by a positive angle  $\alpha_k$  relative to the  $x$ -axis of the NED frame, whose origin is located in  $p_k^n$  and whose the  $x$ -axis is tangential to the path. The position of the USV in the path-fixed frame is computed as

$$[s, e]^T = R_n^p(\alpha_k)(p^n - p_k^n), \quad (1)$$

where  $R_n^p(\alpha_k) \in SO(2)$  is a rotation matrix from the earth-fixed NED frame to the path-fixed frame. As such, the path-fixed  $s$ -coordinate describes the along-track distance, and the  $e$ -coordinate describes the cross-track error. Additional details are found in Fossen (2011, p. 258).

The desired yaw is given by

$$\psi_d = \chi_d - \beta_c, \quad (2)$$

where  $\chi_d$  is the desired course and  $\beta_c = \text{atan2}(v, u) \in \mathbb{S} = (-\pi, \pi)$  is the crab angle caused by currents and wind. Assuming the crab angle is slowly varying, it can be handled with integral action and set to zero (Borhaug et al., 2008). The desired yaw of the vehicle, assuming a lookahead-based LOS guidance system is used, is then given by

$$\psi_d = -\text{atan2}(e, s_\Delta), \quad (3)$$

where  $s_\Delta$  denotes the look-ahead distance to an intersection point  $(x_{\text{los}}, y_{\text{los}})$  on the desired path to  $p_{k+1}^n$  (Borhaug et al., 2008). Assuming an adversary manages to spoof the yaw angle by an offset  $\Delta\psi$ , we have

$$\hat{\psi} = \psi + \Delta\psi, \quad (4)$$

where external disturbances are neglected. The yaw error used by the heading controller is then given by

$$\begin{aligned} \tilde{\psi} &= \psi_d - \hat{\psi} \\ &= \psi_d - \psi - \Delta\psi. \end{aligned} \quad (5)$$

Consequently, the control system steers the yaw to  $\psi = \psi_d - \Delta\psi$  to minimize (5). As such, the USV will pursue a path parallel to the desired path, with a cross-track error given by

$$e_{\Delta\psi} = -s_\Delta \tan \Delta\psi. \quad (6)$$

Hence, we see that adding an offset  $\Delta\psi$  to the yaw results in a predictable change in the USV path. Similarly, if an adversary manages to spoof the position of the vehicle by an offset  $(\Delta x, \Delta y)$ , we have

$$(\hat{x}, \hat{y}) = (x + \Delta x, y + \Delta y), \quad (7)$$

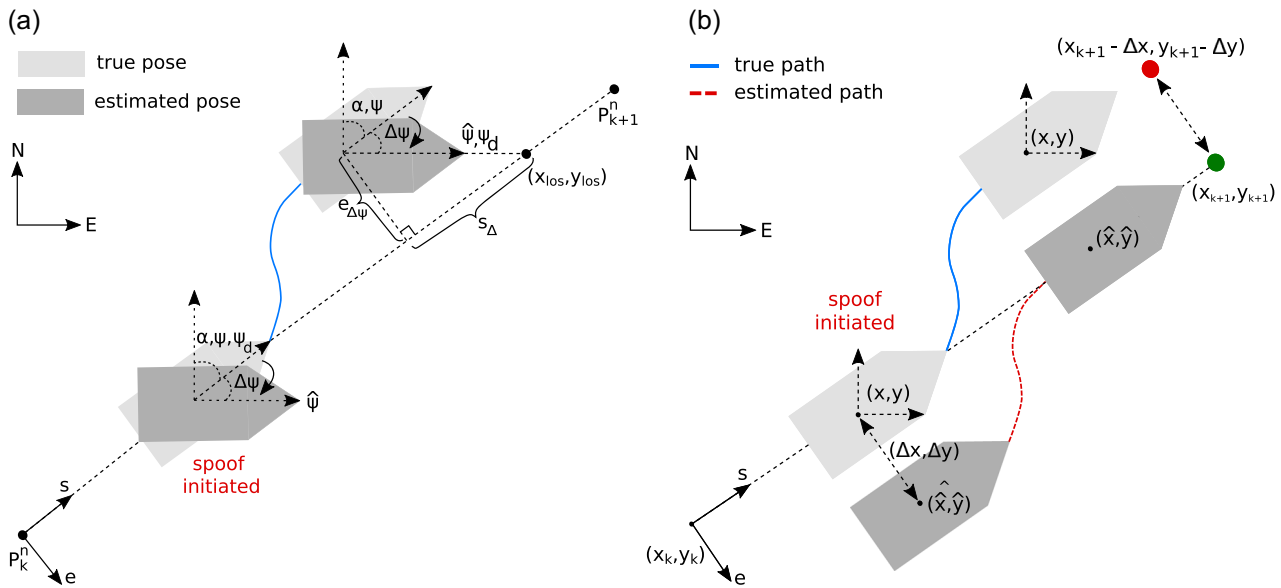
where external disturbances are neglected. Using (1), this translates to offsets in the path-fixed frame as

$$[\Delta s, \Delta e]^T = R_n^p(\alpha_k)[\Delta x, \Delta y]^T. \quad (8)$$

Consequently, assuming a lookahead-based LOS guidance system is used, the guidance system seeks to steer the vehicle towards the desired heading

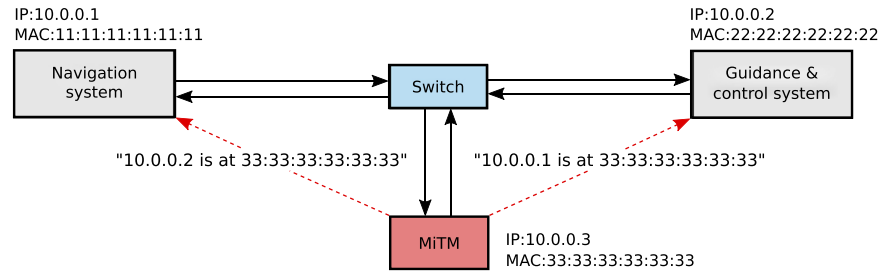
$$\psi_d = -\text{atan2}(e + \Delta e, s_\Delta) \quad (9)$$

sending the vehicle to  $(x_{k+1} - \Delta x, y_{k+1} - \Delta y)$ . Illustrations of the expected behavior when the yaw or the position is spoofed are seen in Figure 3a,b, respectively.



**FIGURE 3** Illustrations of the expected behaviors of the Unmanned Surface Vehicle (USV) when navigation signals are spoofed. External disturbances are neglected. (a) Expected behavior when the yaw of an underactuated USV is spoofed. (b) Expected behavior when the position of an underactuated USV is spoofed [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

**FIGURE 4** A Man-in-The-Middle (MiTM) device that redirects traffic by spoofing the Address Resolution Protocol. IP, Internet Protocol; MAC, message authentication code [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



### 3.3 | Technical implementation

An interesting attack vector against distributed GNC systems communicating over the Internet Protocol (IP) is to redirect the traffic through a device that selectively changes the transmitted data to manipulate the vehicle. For example, the adversary can redirect the traffic by spoofing the Address Resolution Protocol (ARP). The purpose of ARP is to associate an IP address to a link-layer address, and an ARP spoof attack works by falsely associating the link-layer address of the Man-in-The-Middle (MiTM) device with the IP address of the intended recipient. Consequently, the adversary can force all traffic to pass through the MiTM device. An illustration of the attack is shown in Figure 4.

#### 3.3.1 | Injection attack

Assuming a distributed GNC architecture is used, we can connect a single-board computer to an insecure switch and use ARP spoof to redirect the traffic going from the navigation system to the guidance and control system. The computer then runs a script where the contents of the IP packets are analyzed. The IP packets that do not contain the navigation parameter of interest are passed through to the intended recipient, while the IP packets containing the navigation parameter are manipulated. This is possible since the content and the structure of the unencrypted messages are available to the adversary. We use the Python packages `NFQUEUE` (Fox, 2021) and `SCAPY` (Biondi, 2021) to intercept, inspect, manipulate, and retransmit IP packets.

In this case study, the IMC protocol is used to transmit messages. An important observation is that the only integrity check on the IMC messages is a CRC-16 code computed using the generator polynomial  $p(x) = x^{16} \oplus x^{15} \oplus x^2 \oplus 1$  with coefficients in the finite field  $GF(2)$ . Therefore, we can change the message content, after which we forge and append a new, valid CRC-16 code to the message. In Python, CRC functions are readily available using the package `CRCMOD`. Pseudocode describing the MiTM injection attack can be seen in Algorithm 1, where navigation data are manipulated by adding an offset to the navigation parameter.

#### Algorithm 1 Man-in-the-Middle injection attack

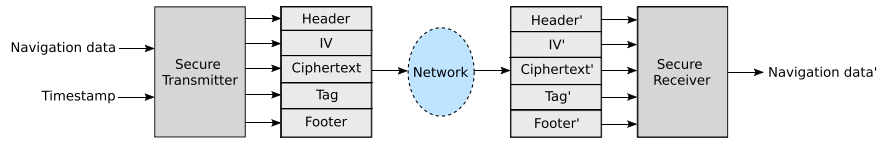
```

 $\theta$ : navigation parameter
 $\Delta_\theta$ : parameter offset
Message: (Header || Payload || Footer)
1: Execute ARP spoof
2: Initialize CRC-16
3: for each intercepted Message do
4:   if Message contains  $\theta$  then
5:      $\theta \leftarrow \text{ReadParameter}(\text{Payload})$ 
6:      $\theta \leftarrow \theta + \Delta_\theta$ 
7:     Payload  $\leftarrow \text{WriteParameter}(\text{Payload}, \theta)$ 
8:     Footer  $\leftarrow \text{CRC-16}(\text{Header} || \text{Payload})$ 
9:     Message  $\leftarrow (\text{Header} || \text{Payload} || \text{Footer})$ 
10:  end if
11:  Transmit(Message)
11: end for

```

#### 3.3.2 | Replay attack

While we can prevent injection attacks with MACs, solely using MACs does not prevent injection of previously transmitted, valid (message, tag)-pairs, and we are therefore vulnerable to *replay attacks*. In a replay attack, a set of authenticated messages can be recorded and later replayed. Since the messages are not changed, the authentication tag is still valid, and the receiver accepts the messages upon reception. Actively steering the vehicle using replayed messages is more challenging since the content of the messages is not necessarily known to the adversary. However, replay attacks can still disrupt the path of the USV. If the messages are encrypted, the data type of the message is more challenging to determine. However, it may still be possible by inspecting the metadata, for example, the size of the packets. Alternatively, all traffic can be logged and replayed. An example of a replay attack is shown in Algorithm 2, where messages are recorded over a predetermined time interval and then immediately replayed.



**FIGURE 5** A flowchart of secured communication between the navigation system and the guidance and control system. IV, Initialization Vector [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

#### Algorithm 2 Man-in-the-Middle replay attack

$\theta$ : navigation parameter  
 $t$ : time since initialization  
 $\tau$ : duration of replay attack  
 Message: (Header || Payload || Footer)  
 Q: queue for messages

- 1: Execute ARP spoof
- 2: **for** each intercepted Message
- 3:   **if** Message contains  $\theta$  **and**  $t < \tau$  **then**
- 4:     Q.enqueue(Message)
- 5:   **else if** Message contains  $\theta$  **then**
- 6:     Message  $\leftarrow$  Q.dequeue()
- 7:   **end if**
- 8:   Transmit(Message)
- 9: **end for**

#### 2: **While true do**

- 3:   Initialize  $AE_{K,IV}$
- 4:   Instantiate SecureMessage
- 5:   NavigationData  $\leftarrow$  ReadNavigationData()
- 6:   T  $\leftarrow$  GetTime()
- 7:   (Header || IV || Ciphertext || Tag)  $\leftarrow$   $AE_{K,IV}$ (Header || T || NavigationData)
- 8:   Footer  $\leftarrow$  CRC-16(Header || IV || Ciphertext || Tag)
- 9:   SecureMessage  $\leftarrow$  (Header || IV || Ciphertext || Tag || Footer)
- 10:   Transmit(SecureMessage)
- 11:   Update IV
- 12: **end while**

### 3.3.3 | Securing the navigation data

To secure the navigation data against injection and replay attacks, we can use authenticated encryption with the addition of timestamps or sequence numbers. In our example, we add a fresh timestamp to the navigation data before both are encrypted. We then compute a MAC tag over the resulting ciphertext, the header of the message, and the IV. Upon reception, we recompute the MAC tag and decrypt the navigation data and the timestamp. If the recomputed and received tags match and the timestamp is fresh, the navigation data are accepted. An illustration of the signal flow with the proposed secure transmission and reception algorithms is shown in Figure 5, and pseudocodes for the secure transmitter and receiver are found in Algorithms 3 and 4, respectively. We use the authenticated encryption algorithm AEGIS, a cryptographically strong authenticated encryption algorithm that has been shown to provide excellent performance in software with negligible time delays (Volden et al., 2021). The AEGIS implementation used is publicly available and described by Solnør (2020).

#### Algorithm 3 Secure transmitter

K: Symmetric key; IV: Initialization Vector;  
 $AE_{K,IV}$ : Authenticated encryption function parameterized by K and IV;  
 SecureMessage: (Header || Payload || Footer)  
 T: Timestamp

- 1: Initialize CRC-16

#### Algorithm 4 Secure receiver

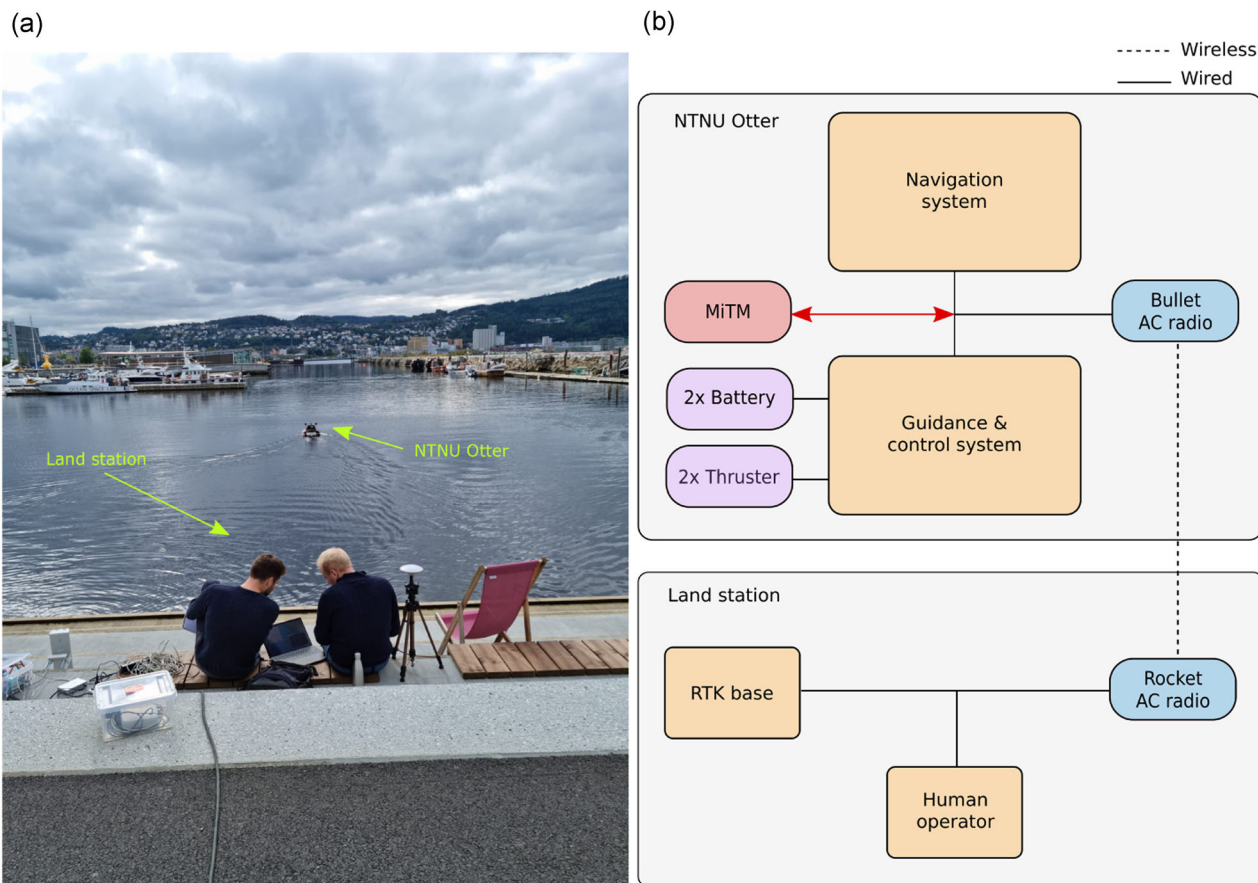
K: Symmetric key; IV: Initialization Vector;  
 $AD_{K,IV}$ : Authenticated decryption function parameterized by K and IV;  
 SecureMessage: (Header || Payload || Footer)  
 T: Timestamp

- 1: T = 0
- 2: **for** each received SecureMessage **do**
- 3:   (Header' || IV' || Ciphertext' || Tag' || Footer')  $\leftarrow$  Read (SecureMessage)
- 4:   Initialize  $AD_{K,IV}'$
- 5:   (T || NavigationData || Tag)  $\leftarrow$   $AD_{K,IV}'$ (Header' || IV' || Ciphertext' || Tag')
- 6:   **if** Tag == Tag' **and** T' > T **then**
- 7:     T  $\leftarrow$  T'
- 8:     Accept NavigationData'
- 9:   **else**
- 10:     Reject NavigationData'
- 11:   **end if**
- 12: **end for**

## 4 | EXPERIMENTAL SETUP

The experimental setup consists of the NTNU Otter USV and a land station. The NTNU Otter uses a distributed GNC system in which the navigation and guidance and control system are two separate systems that communicate over Ethernet. Furthermore, we assume that an adversary has gained access to the signal transmission onboard the NTNU Otter between the navigation and guidance and control system. The land station consists of a Real-Time Kinematic (RTK) base station that sends correction data to the navigation system and a remote laptop for the operator to





**FIGURE 6** (a) An overview of the experimental scene showing the base station and the NTNU Otter Unmanned Surface Vehicle (USV). (b) A high-level schematic of the land station and the USV. MiTM, Man-in-The-Middle; RTK, Real-Time Kinematic [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

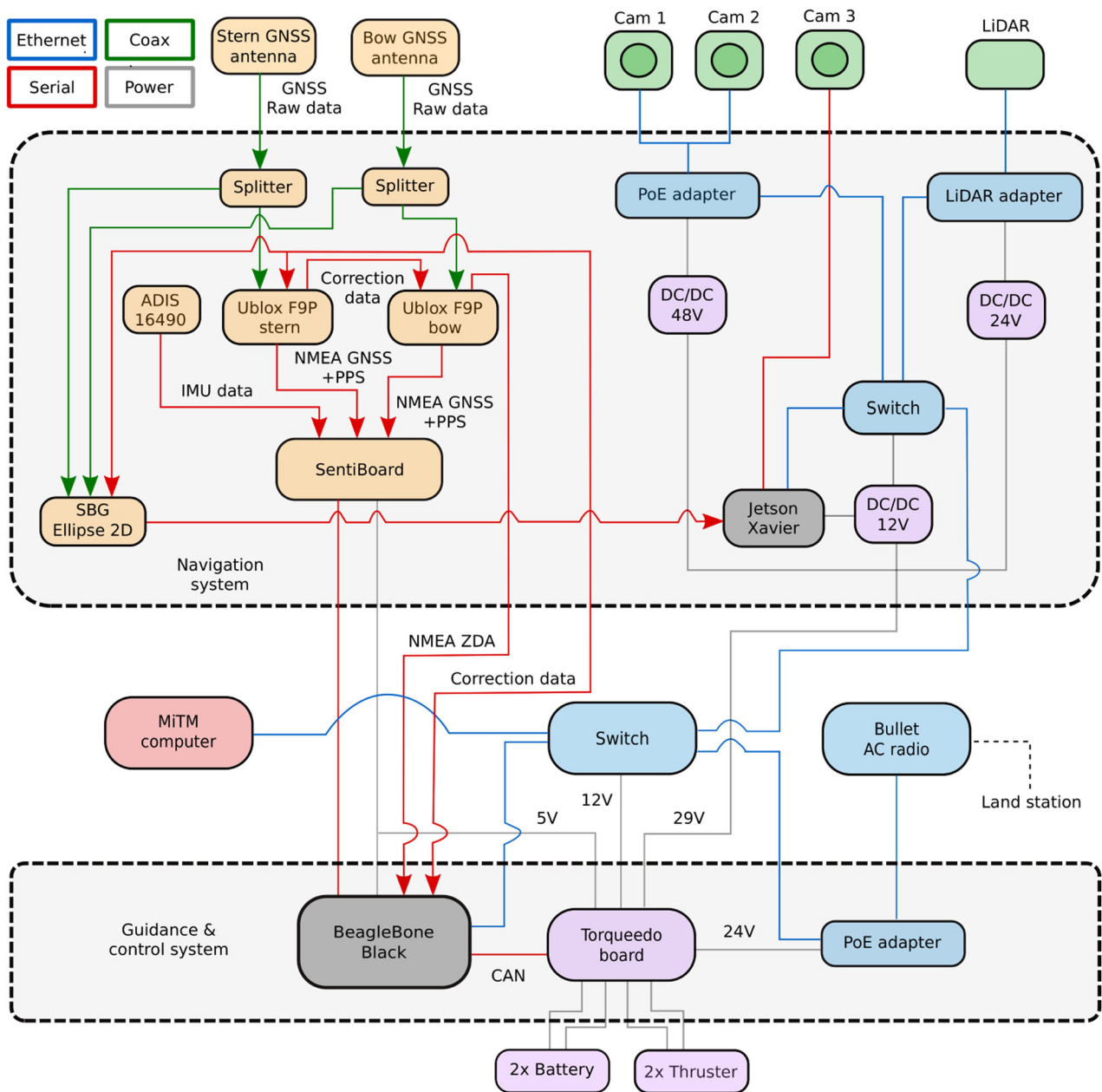
upload missions or control the USV directly. The land station and the NTNU Otter communicates using a point-to-point transparent Ethernet bridge established over radio communication. Figure 6a shows an overview of the experimental scene, and Figure 6b shows a schematic of the experimental setup.

## 4.1 | The NTNU Otter

The NTNU Otter is underactuated with fixed starboard and port thrusters mounted at the stern. The software and hardware architectures were designed and built at the Department of Engineering Cybernetics, NTNU, while the body, thrusters, batteries, and the power interface board were purchased from Maritime Robotics AS. A schematic of the hardware onboard the NTNU Otter is shown in Figure 7. We use the LSTS software toolchain, consisting of DUNE, the IMC protocol, and the Neptus Graphical User Interface (GUI), to control and interact with the vehicle. DUNE is used for guidance, control, and navigation and to interface with hardware components, while the IMC protocol is used to transmit data between individual DUNE tasks. Finally, we use the Neptus GUI to interact with the vehicle by passing maneuvers to the guidance system or remote controlling the USV from the land station.

### 4.1.1 | Navigation system

The NTNU Otter uses two independent navigation systems. The first navigation system consists of an ADIS 16490 IMU (Analog Devices, 2021) and two U-blox F9P GNSS receivers (U-blox, 2021) with synchronized data acquisition through a SentiBoard (Senti Systems, 2021). The first GNSS receiver is configured as a “moving base” and receives raw GNSS data from an antenna mounted at the stern of the NTNU Otter and correction data from the RTK base. The second GNSS receiver is configured as a “rover” and receives raw GNSS data from an antenna mounted at the bow of the NTNU Otter and correction data from the moving base. As such, the rover finds the yaw of the USV. The second navigation system consists of an SBG Ellipse 2D INS (SBG Systems, 2021), which receives raw GNSS data from the stern and bow antennas and correction data from the base station. For our experiments, the navigation data from the SBG Ellipse 2D was used in feedback control, while the navigation data from the SentiBoard was used as ground truth measurements for comparison. Since the navigation systems receive corrections from the same base with centimeter precision, the navigation data produced are almost identical. As such, the effect of measurement noise is reduced to a minimum. The navigation system also contains vision-based sensors,



**FIGURE 7** A hardware schematic of the navigation system and guidance and control system components of the NTNU Otter unmanned surface vehicle. DC, direct current; GNSS, Global Navigation Satellite System; IMU, Inertial Measurement Unit; MiTM, Man-in-The-Middle; NMEA, National Marine Electronics Association; PoE, Power over Ethernet; PPS, pulse per second; RTK, Real-Time Kinematic [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

that is, cameras and a LiDAR, that can be used for local navigation purposes. However, these sensors are not used for the experiments. A schematic of the complete navigation system can be seen in the upper part of Figure 7.

#### 4.1.2 | Guidance system

The guidance system consists of a path planner and a LOS guidance law with integral action (ILOS) (Caharija et al., 2012). The guidance

system receives a set of WPs and desired speeds from the operator, and the estimated position, velocity, and yaw, from the navigation system. The path planner then produces the desired path using the WPs and the desired speeds. Then, the ILOS guidance law computes the desired yaw based on the estimated state and the desired path. Using the following condition

$$\frac{\sqrt{\left(\|IR_n^p (p_{k+1}^n - p_k^n)\|_2 - s\right)^2 + e^2}}{u} - C_t \leq 0, \quad (10)$$

the path planner determines whether a WP has been reached or not. Here,  $C_t$  is a positive constant. To avoid problems with integral windup resulting in large overshoots, the integral action of the ILOS guidance law is only used when the USV is located within a certain distance from the desired path (Caharija, 2014). In practice, we use a cross-track distance of 2.5 m to determine whether integral action is enabled or not.

#### 4.1.3 | Control system

The control system consists of a proportional-integral speed controller and a proportional heading controller. On the basis of the estimated state from the navigation system and the desired speed and yaw from the guidance system, the control system produces desired revolutions per minute of the starboard and port thrusters. The speed controller contains logic that disables the controller if the difference between desired and estimated yaw exceeds  $36^\circ$  to reduce the cross-track error following sharp turns. The control system also permits remote operation, in which manual control signals can be transmitted from a PlayStation 4 (PS4) controller connected to the remote control laptop. An illustration of the signal flow of the closed-loop system is shown in Figure 8.

#### 4.1.4 | Synchronization

We use the Precision Time Protocol (PTP) to synchronize the hardware clocks onboard the NTNU Otter. With PTP, the devices are synchronized with sub-microsecond precision using a master-slave setup (Chaloupka et al., 2015). We configure the Beaglebone Black computer (Kridner et al., 2021) in the guidance and control system to be the master clock, and we configure the Jetson Xavier computer (Nvidia, 2021) in the navigation system to be the slave clock. The master clock derives the time from a GNSS receiver using the NMEA ZDA message, as shown in Figure 7. Furthermore, we use a SentiBoard for data synchronization. The SentiBoard is synchronized with Coordinated Universal Time (UTC) using a time-of-validity (TOV)

signal, often referred to as the pulse per second, from the GNSS receivers. The IMU also produces a TOV each cycle, after which the SentiBoard reads and timestamps the IMU data, in hardware, with its internal clock. With this setup, the data are synchronized to UTC with a root-mean-squared clock drift of  $1.9 \mu\text{s/s}$  (Albrektsen, 2018).

## 4.2 | Land station

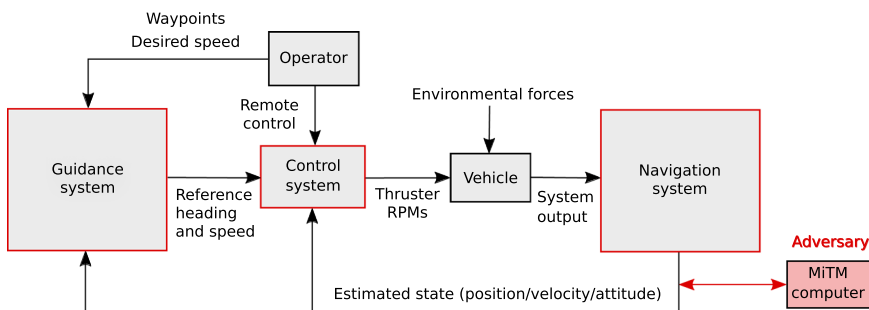
The land station consists of a remote control computer running the Neptus GUI and an RTK base station that transmits corrections to the navigation system. We used the remote control computer to create and upload missions to the guidance system or control the vehicle manually with a PS4 controller. The RTK base station consists of a GNSS antenna, a U-blox F9P GNSS receiver, and a Beaglebone Black, as shown in Figure 9. We configured the GNSS receiver to estimate the phase of the GNSS carrier wave over 17 h before we conducted the experiments. This surveying procedure resulted in an absolute precision of 6 cm, negatively affected by a cruise ship that docked close to the GNSS antenna during the survey.

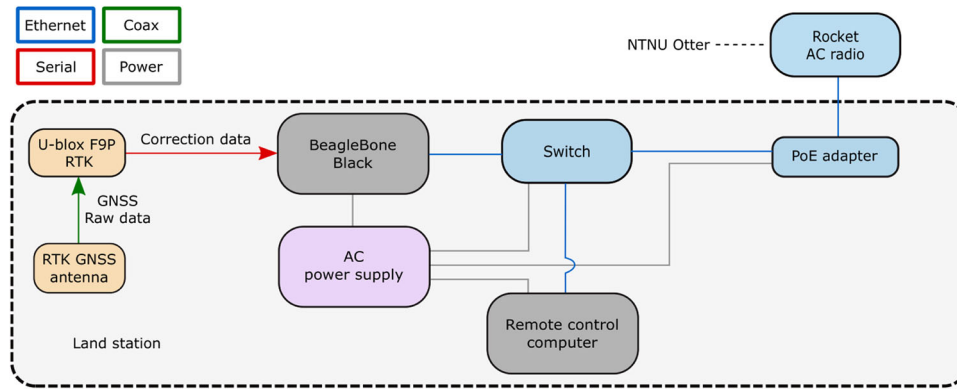
## 4.3 | Experimental description

We perform five field experiments to demonstrate the vulnerability of the distributed GNC system onboard the USV in the harbor environment. The desired paths of the vehicle during the experiments are shown in Figure 10. Experiments 1–4 are conducted with desired path 1, where the desired speed between WP<sub>1</sub> and WP<sub>2</sub> is set to 0.5 and 0.25 m/s in Experiments 1 and 2 and Experiments 3 and 4, respectively. Experiment 5 was conducted using desired path 2 with desired speed set to 0.5 m/s between the WPs.

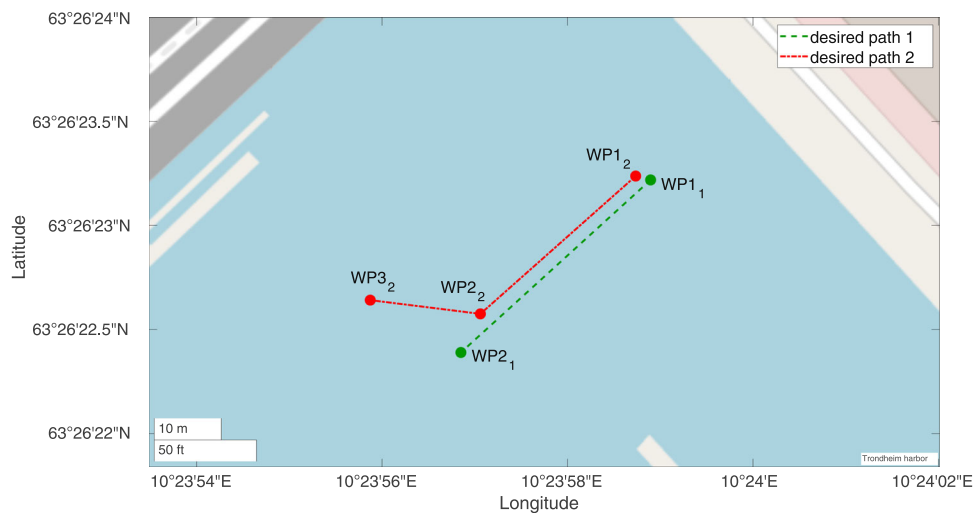
We manipulate the vehicle by adding fixed offsets to the yaw and latitude estimates in Experiments 1 and 2, respectively. In Experiment 1, we alter the heading by adding a fixed offset of  $57.3^\circ$ , and in Experiment 2, we change the latitude by adding a fixed offset of approximately 10 m. Since large offsets are easy to detect, we also implement attacks where the yaw and latitude are changed by incremental offsets, slowly dragging the vehicle off course. Consequently, we manipulate the

**FIGURE 8** A closed-loop guidance, navigation, and control system of the NTNU Otter unmanned surface vehicle under attack by a Man-in-The-Middle (MiTM) adversary [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]





**FIGURE 9** A hardware schematic of the land station components. AC, alternating current; GNSS, Global Navigation Satellite System; RTK, Real-Time Kinematic [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



**FIGURE 10** Predefined waypoints (WPs) determine the desired paths of the vehicle used in the experiments [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

vehicle by adding incremental offsets of  $0.573^\circ/s$  and  $0.07\text{ m/s}$  to the yaw and the latitude in Experiments 3 and 4, respectively. The speed was lowered to  $0.25\text{ m/s}$  for the incremental spoofing attacks to take effect over an extended period. In Experiments 1–4, we initiate the attacks when the vehicle is between  $WP1_1$  and  $WP2_1$ . We proceed by performing a replay attack in Experiment 5, where a sequence of encrypted and authenticated messages containing heading information from the navigation system is recorded and replayed with a 30-s delay to manipulate the vehicle. The vehicle heading is recorded between  $WP1_2$  and  $WP2_2$  and replayed just before the planned course change at  $WP2_2$ . We use the second path in this experiment to see how the vehicle handles the planned course change while receiving delayed heading information.

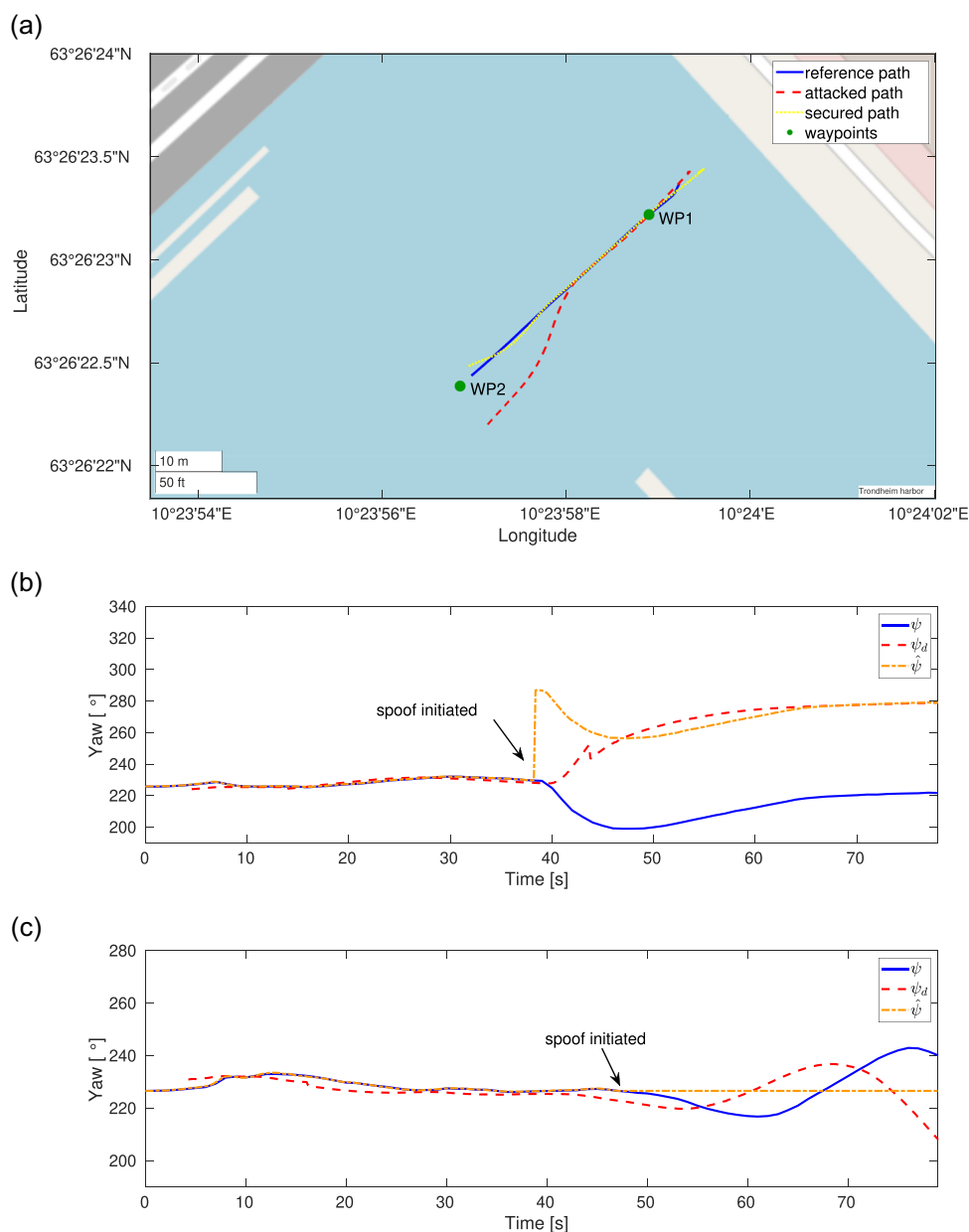
We include three scenarios for each experiment. First, we execute a reference scenario to observe how well the vehicle follows the path while affected by environmental forces, such as winds and currents. We then perform an attack scenario to show how the USV is affected by the attack. Finally, we execute a secured scenario to see how well the added countermeasures protect the vehicle against the attacks.

## 5 | EXPERIMENTAL RESULTS

We present the results of the experiments by plotting the USV position during the attack scenario against the position of the vehicle in the reference scenario and the secured scenario. The manipulated parameter, that is, heading or position, is plotted against the true value of the parameter obtained by the redundant navigation system. When the heading is spoofed, we also plot the desired heading from the guidance system. At last, we show the effect of using the proposed secure transmitter and receiver, described in Algorithms 3 and 4.

### 5.1 | Experiment 1: Fixed heading spoof

The results from Experiment 1 are shown in Figure 11. Figure 11a shows how the vehicle deviates from the desired path, and Figure 11b shows how the estimated heading changes after adding a fixed heading offset. When we secure the signal transmission



**FIGURE 11** The results from Experiment 1. (a) The paths between the waypoints (WPs) of the Unmanned Surface Vehicle (USV) in the three scenarios. (b) True, desired, and estimated heading of the USV when we attack the insecure system with a fixed heading offset. (c) True, desired, and estimated heading of the USV when we attack the secured system with a fixed heading offset [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

with authenticated encryption, the spoofing attack is detected immediately, and all spoofed messages are dismissed. The control system uses the latest heading estimate available before the attack. As a result, the vehicle continues along the desired path with an oscillating heading, as shown in Figure 11c.

## 5.2 | Experiment 2: Fixed latitude spoof

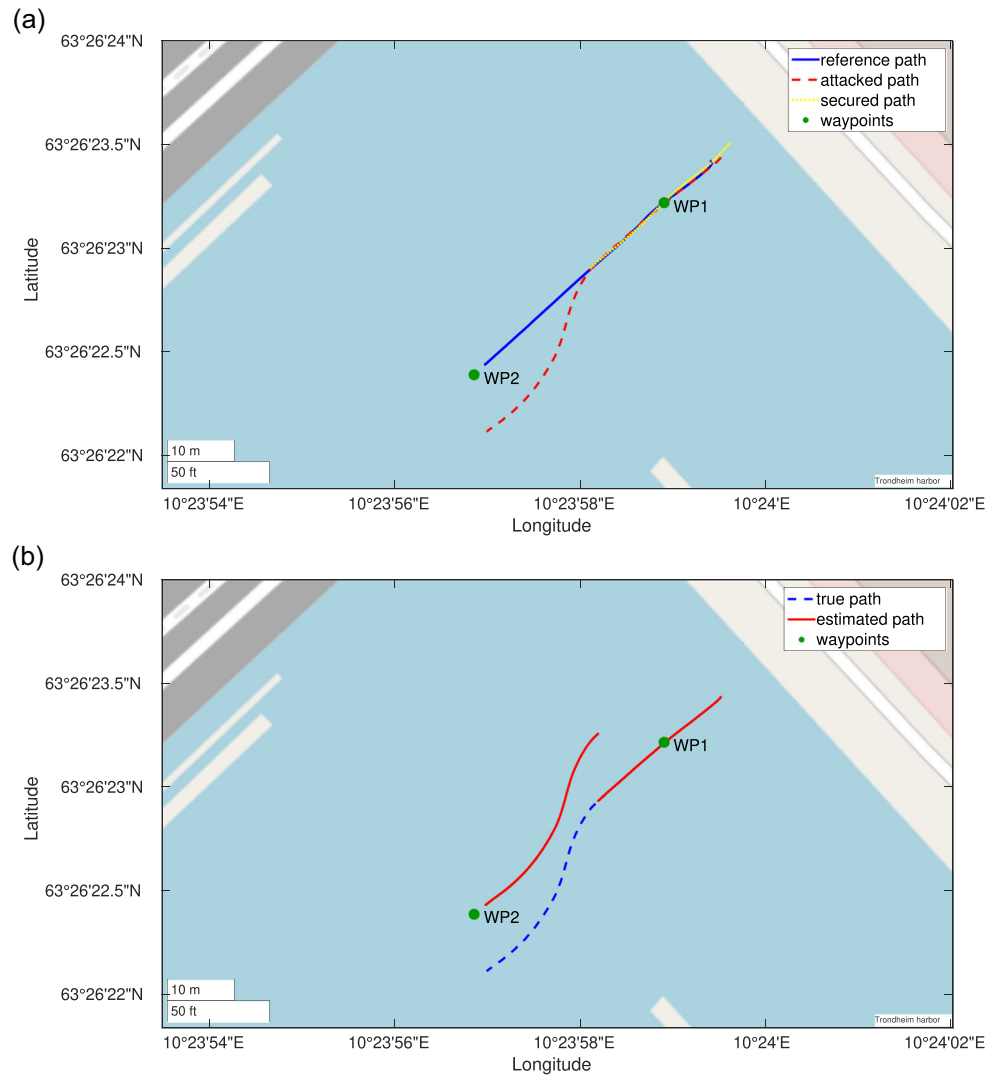
The results from Experiment 2 are shown in Figure 12. Figure 12a shows how the fixed latitude spoof successfully puts the vehicle off course. In Figure 12, we plot the true and the

estimated paths of the USV during the attack scenario, showing the sudden jump in the estimated position when we launch the attack. When we secure the signal transmission with authenticated encryption, the spoofing attack is detected, and all spoofed messages are dismissed. Without updated position estimates, the vehicle goes to an error state, and the mission is aborted.

## 5.3 | Experiment 3: Incremental heading spoof

The results from Experiment 3 are shown in Figure 13. Figure 13a shows the paths of the vehicle in the three scenarios. The effect of





**FIGURE 12** The results from Experiment 2. (a) The paths between the waypoints (WPs) of the Unmanned Surface Vehicle (USV) in the three scenarios. (b) True and estimated path of the USV when we attack the insecure system with a fixed latitude offset [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

the incremental attack is not immediately visible on the path of the vehicle in the attack scenario. However, the vehicle veers off course in an attempt to correct its heading towards the end. The increasing deviation between the true and estimated heading of the vehicle is visible in Figure 13b. When the signal transmission is secured with authenticated encryption, the spoofing attack is detected and all spoofed messages are dismissed. Similar to Experiment 1, the vessel continues along its desired path; however, the heading oscillations are more pronounced because the USV operates without an updated heading estimate for an extended time period, as can be seen in Figure 13c.

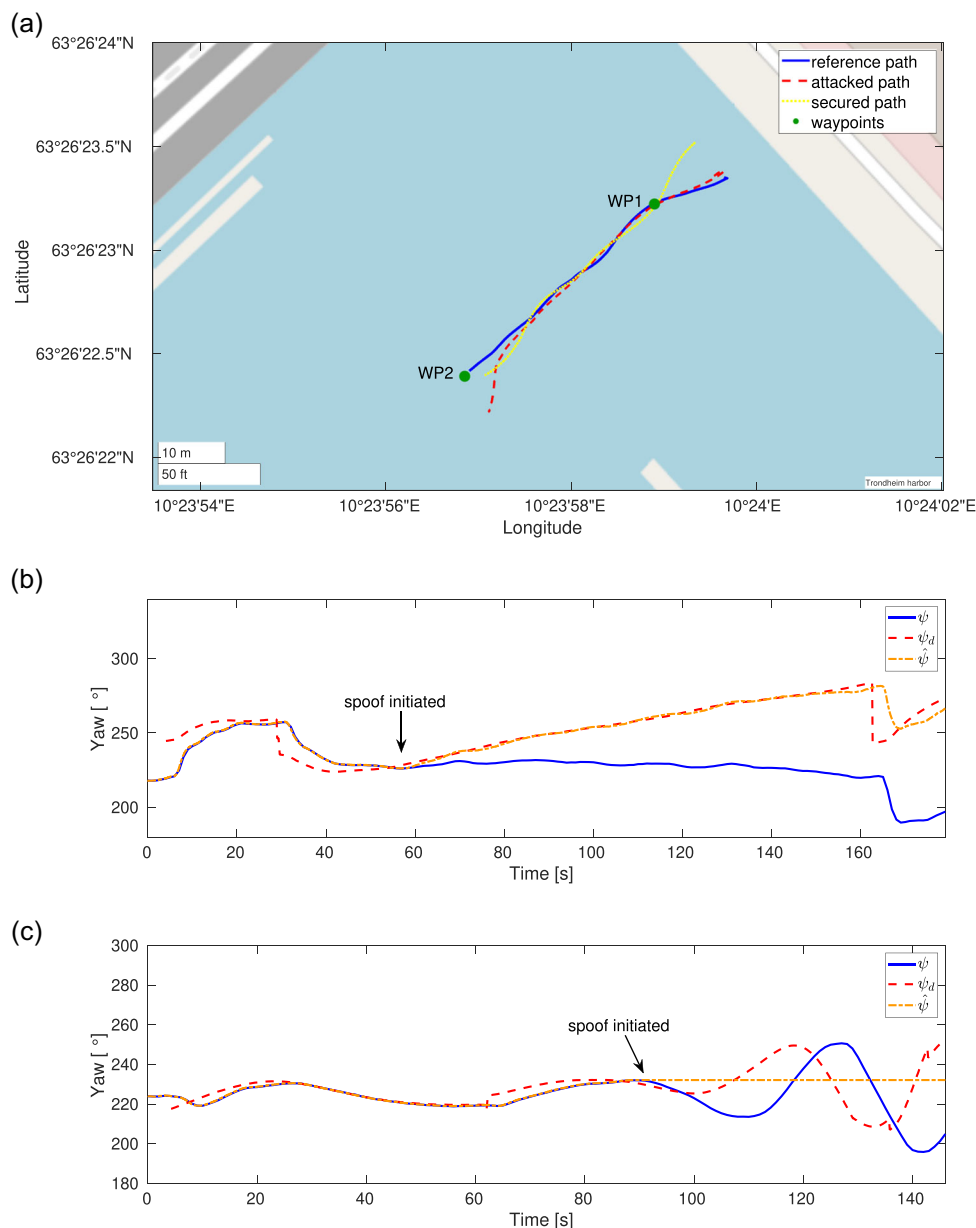
#### 5.4 | Experiment 4: Incremental latitude spoof

The results from Experiment 4 are shown in Figure 14. Figure 14a shows the paths of the vehicle in the three scenarios. We

successfully drag the USV off course in the attack scenario by adding an incremental offset to the latitude estimate. We show this in Figure 14b, where we plot the true and the estimated path of the USV. When we secure the system using authenticated encryption, the spoofed messages are dismissed, and the vehicle enters an error state. Consequently, the mission is aborted.

#### 5.5 | Experiment 5: Replay attack

The results from Experiment 5 are shown in Figure 15. Figure 15a shows the paths of the vehicle in the three scenarios. The replay attack is seen to cause a slightly delayed action compared with the reference path. Furthermore, Figure 15b shows that the replay attack successfully changes the estimated heading immediately before the USV reaches WP<sub>2</sub>. When we secure the system by adding authenticated timestamps, the replayed messages are identified and



**FIGURE 13** The results from Experiment 3. (a) The paths between the waypoints (WPs) of the Unmanned Surface Vehicle (USV) in the three scenarios. (b) True, desired, and estimated heading of the USV when we attack the insecure system with an incremental heading offset. (c) True, desired, and estimated heading of the USV when we attack the secured system with an incremental heading offset [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

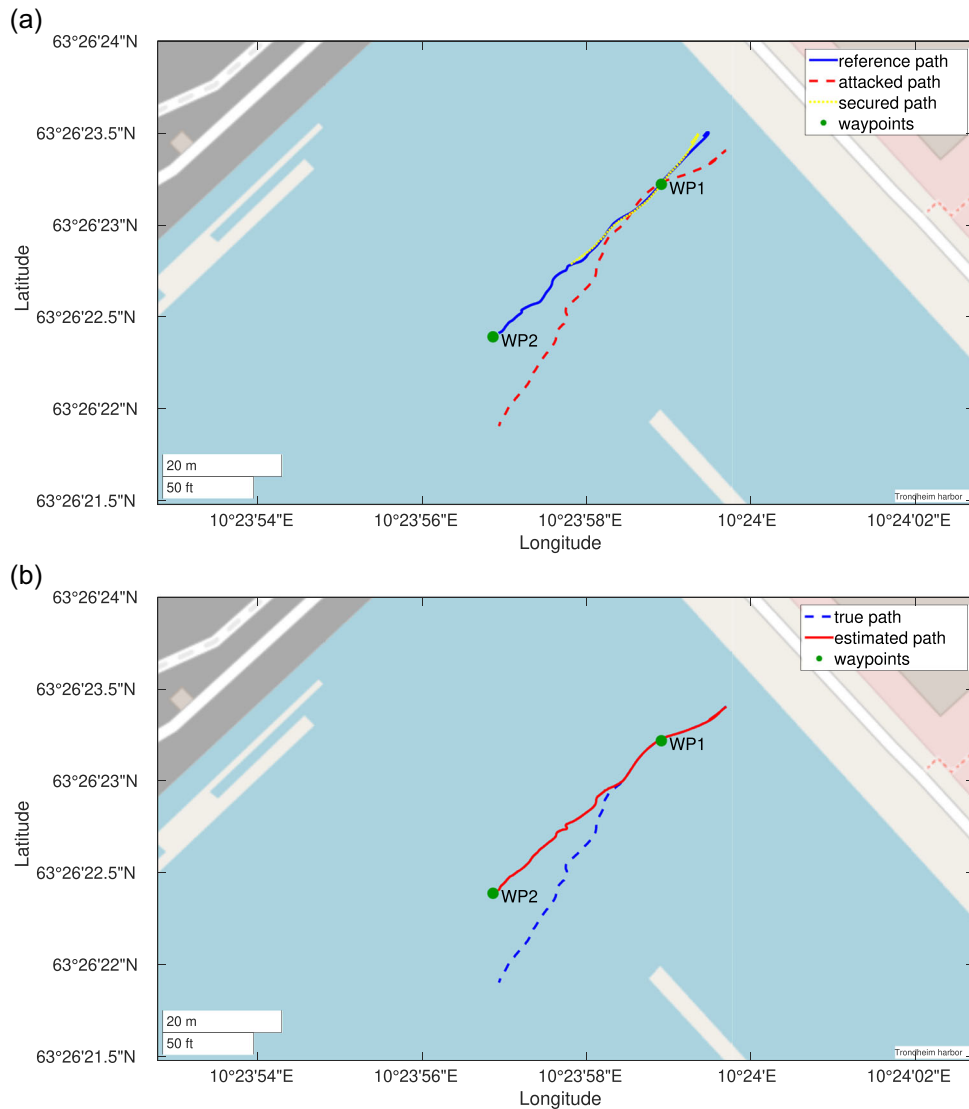
discarded. Consequently, the vehicle enters an error state, and the mission is aborted.

## 5.6 | Discussion of results

The experiments demonstrated that ARP spooft is an effective attack vector against distributed GNC systems communicating over a local network. Furthermore, they showed that messages transmitted using protocols merely relying on conventional fault checks to detect invalid messages are vulnerable to eavesdropping and injection attacks. As

expected, manipulation of heading and position estimates caused a predictable change in the path of the underactuated USV. Notice that, while we here considered attacks where the spoofed values were computed by adding offsets, more advanced methods of selecting the spoofed values can be used without fundamentally changing the attack. Furthermore, we demonstrated that authentication is insufficient to prevent the successful injection of recorded messages through a replay attack. Finally, we showed that authenticated encryption with timestamps effectively prevents the hijacking of the USV.

When the attacks caused the USV to deviate from the desired path, the integral effect of the ILOS guidance was switched off when



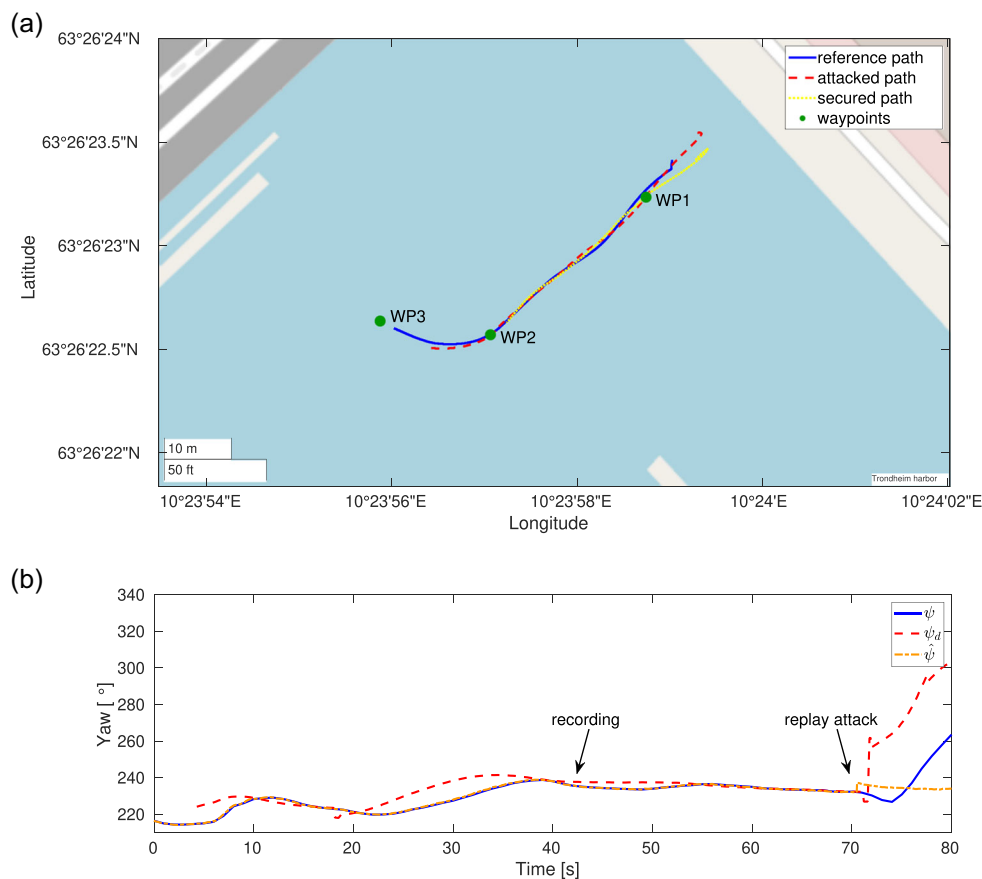
**FIGURE 14** The results from Experiment 4. (a) The paths between the waypoints (WPs) of the Unmanned Surface Vehicle (USV) in the three scenarios. (b) True and estimated path of the USV when we attack the insecure system with an incremental latitude offset [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

the cross-track distance exceeded the 2.5 m threshold. We see this in Figures 11b and 13b, where we observe sudden jumps in the desired heading when the heading was spoofed. The effect is especially pronounced in the incremental heading spoof. We believe this is because the integral action had been enabled over an extended period, and the weather conditions were worsening with strong wind gusts and currents, causing a varying crab angle. The varying crab angle also strongly influenced the incremental latitude spoof, with the USV oscillating around the desired path, as seen in Figure 14a. It is clear that the assumption that the crab angle would be slowly varying, used by the ILOS guidance law, was not satisfied during these experiments.

To secure the USV against the hijacking attempts demonstrated in Experiments 1–4, Algorithms 3 and 4 were used by the navigation system and guidance and control systems, respectively. When an attack was detected, we used two separate failure modes for position

spoofs and heading spoofs. When a position spoof was detected, the USV went to a failure mode and halted all actions when no recent valid position estimates were available. In contrast, heading spoofs were handled by using the most recent, valid heading estimate available. Notably, the latter resulted in oscillating behavior of the USV, as seen in Figures 11a,c and 13a,c. Consequently, we found that if a heading spoof is detected and no new heading estimates are available, the USV should, instead, go to a failure mode to prevent erratic behavior. We believe that more extensive safety analysis, and the development of relevant failure modes, are important steps towards safe autonomous vehicles.

As shown from the desired and the true heading in Figure 15b, the replay attack resulted in delayed action of the heading controller. This delay resulted in a slight change of the path. From Figure 15a, we also observed that the mission was fulfilled approximately 8 m before WP<sub>2</sub>. Because of the planned course change in WP<sub>2</sub>, the



**FIGURE 15** The results from Experiment 5. (a) The paths between the waypoints (WPs) of the Unmanned Surface Vehicle (USV) in the three scenarios. (b) True, desired, and estimated heading of the USV when we attack the insecure system with a replay attack [Color figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

difference between the desired and the estimated heading exceeded a  $36^\circ$  threshold, at which point the speed controller was switched off. This resulted in a significant increase in surge speed  $u$ . Consequently, because of the increase in surge speed, the inequality (10) was satisfied early, and the path planner prematurely announced that the mission had been completed. In the reference path, the speed controller largely remained on, and the USV got much closer to WP3<sub>2</sub> before the path planner announced that the mission had been completed. Unfortunately, the distance between WP2<sub>2</sub> and WP3<sub>2</sub> was not sufficiently large to fully capture the consequence of the attack. Nevertheless, the attack successfully changed the estimated heading of the USV. When the communication was secured using Algorithms 3 and 4, the replay attack was immediately detected. When messages with old timestamps were detected, and no fresh heading estimates were available, the USV aborted the mission and went into an error state instead of continuing along the desired path.

It is clear that when Algorithms 3 and 4 were used, the attacks still managed to take the USV out of service. However, when an adversary gains access to the transmission lines of the GNC system, DoS attacks are trivial to execute. Additionally, the proposed algorithms do not prevent delay attacks where the MiTM device merely delays messages instead of replaying them. However, actively

steering the vehicle through such an attack with encrypted messages is highly unlikely since the device has no means of knowing the contents of the delayed messages. Consequently, we classify this as a DoS attack. Possible methods to detect such attacks range from comparing the interval between received messages to an expected value and comparing timestamps on received messages to the local clock. Importantly, keeping the USV in service should not be the goal. Instead, the important takeaway is that spoofed and replayed messages are detected and discarded such that the vehicle cannot actively be steered, that is, hijacked, by the adversary.

## 6 | CONCLUSIONS

With recent advances in ICT paving the way for increased use of unmanned and autonomous vehicles, implementing secure GNC systems is crucial to ensure safe and reliable operation. Successful cyber-attacks, for example, as part of a terrorist attack, may cause fatal human and financial consequences and devastate trust by authorities, investors, and the general public. Previous studies have highlighted potential vulnerabilities in autonomous vehicles through surveys and high-level studies. Among the few studies demonstrating

attacks against intravehicular communication, experimental verification has been limited to conventional vehicles and controlled laboratory environments. Hence, there is a gap between theory and practice in cybersecurity for autonomous vehicles. Furthermore, studies presenting countermeasures usually resort to anomaly-based IDSs. However, anomaly-based IDSs suffer from high false-positive rates, and because of the base rate fallacy, these systems are not appropriate for practical applications.

In this paper, we have addressed these problems and verified and analyzed the effects of the proposed attacks and countermeasures through field experiments. First, we have demonstrated how injection attacks can actively take control of an underactuated USV, thus bridging the gap between theory and practice. Second, we have shown how cryptographic methods effectively prevent attacks against intravehicular communication. Consequently, we recommend that developers actively secure intravehicular communication in GNC systems by using the proposed secure transmitter and receiver algorithms combining authenticated encryption, for example, the AEGIS framework, with additional plaintext redundancy, such as timestamps or sequence numbers, to prevent eavesdropping, injection, and replay attacks.

## ACKNOWLEDGMENTS

This study was funded by the Research Council of Norway (project no. 223254) through the NTNU Center of Autonomous Marine Operations and Systems (AMOS) at the Norwegian University of Science and Technology.

## DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

## ORCID

Petter Solnør  <https://orcid.org/0000-0001-9885-9662>

Øystein Volden  <https://orcid.org/0000-0002-6623-9036>

Kristoffer Gryte  <https://orcid.org/0000-0002-2223-4129>

Slobodan Petrovic  <https://orcid.org/0000-0002-4435-2716>

Thor I. Fossen  <https://orcid.org/0000-0003-0911-7021>

## REFERENCES

- ABI Research. (2019) *The rise of ROS: Nearly 55% of total commercial robots shipped in 2024 will have at least one Robot Operating System package installed*. Business Wire, New York, New York, USA.
- Albrektsen, S.M. (2018) *Sensor synchronization and navigation in GNSS-denied environments for unmanned aerial vehicles*. Ph.D. Thesis, Norwegian University of Science and Technology.
- Analog Devices. (2021) ADIS16490. Available at: <https://www.analog.com/media/en/technical-documentation/data-sheets/adis16490.pdf> [Accessed: 2021-10-26].
- Axelsson, S. (2000) The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security*, 3(3), 186–205.
- Barker, E. & Roginsky, A. (2019) *NIST special publication 800-131A—transitioning the use of cryptographic algorithms and key lengths*. Revision 2. NIST Technical Series Publications.
- Bellare, M. & Namprempre, C. (2008) Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *The Journal of Cryptology*, 21(4), 469–491.
- Biondi, P. (2021) *Scapy*. Available at: <https://scapy.net> [Accessed: 2021-10-22].
- Bolbot, V., Theotokatos, G., Boulougouris, E. & Vassalos, D. (2020) A novel cyber-risk assessment method for ship systems. *Safety Science*, 131, 104908.
- Borhaug, E., Pavlov, A. & Pettersen, K.Y. (2008) Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents. In: 2008 47<sup>th</sup> IEEE conference on decision and control, pp. 4984–4991. Cancun, Mexico: IEEE.
- Caharija, W. (2014) *Integral line-of-sight guidance and control of underactuated marine vehicles*. Ph.D. Thesis, Norwegian University of Science and Technology.
- Caharija, W., Candeloro, M., Pettersen, K.Y. & Sørensen, A.J. (2012) Relative velocity control and integral LOS for path following of underactuated surface vessels. In: Bruzzone G. & Caccia M. (Eds.) *IFAC proceedings of the 9<sup>th</sup> IFAC conference on manoeuvring and control of marine craft*. Vol. 45(27), pp. 380–385. Arezano, Italy: Elsevier Limited.
- Chaloupka, Z., Alsindi, N. & Aweya, J. (2015) Transparent clock characterization using IEEE 1588 PTP timestamping probe. In: *Proceedings of the 2015 IEEE international instrumentation and measurement technology conference (I2MTC)*, pp. 1537–1542. Pisa, Italy: IEEE.
- Dang, Q.H. (2008) *The keyed-hash message authentication code (HMAC)—FIPS 198-1*. Gaithersburg, MD: National Institute of Standards and Technology. Technical Report.
- Dieber, B., White, R., Taurer, S., Breiling, B., Caiazza, G., Christensen, H. & Cortesi, A. (2020) Penetration testing ROS. In: Koubaa, A. (Ed.), *Robot operating system (ROS): The complete reference*, Vol. 4. Cham: Springer International Publishing, pp. 183–225.
- Dzung, D., Naedele, M., VonHoff, T. & Crevatin, M. (2005) Security for industrial communication systems. *Proceedings of the IEEE*, 93(6), 1152–1177.
- El-Rewini, Z., Sadatsharan, K., Selvaraj, D.F., Plathottam, S.J. & Ranganathan, P. (2020) Cybersecurity challenges in vehicular communications. *Vehicular Communications*, 23, 100214.
- Fazzari, K. (2021) *ROS 2 DDS-security integration*. Available at: [https://design.ros2.org/articles/ros2\\_dds\\_security](https://design.ros2.org/articles/ros2_dds_security) [Accessed: 2021-09-29].
- Felski, A. & Zwolak, K. (2020) The ocean-going autonomous ship-challenges and threats. *Journal of Marine Science and Engineering*, 8(1), 41–56.
- Fossen, T.I. (2011) *Handbook of marine craft hydrodynamics and motion control*, 1st edition. Chichester, West Sussex, United Kingdom: Wiley.
- Fossen, T.I. (2021) *Handbook of marine craft hydrodynamics and motion control*, 2nd edition. Chichester, West Sussex, United Kingdom: Wiley.
- Fox, M. (2021) *NetfilerQueue*. Available at: <https://pypi.org/project/NetfilerQueue/> [Accessed: 2021-10-25].
- Han, M.L., Kwak, B.I. & Kim, H.K. (2021) Event-triggered interval-based anomaly detection and attack identification methods for an in-vehicle network. *IEEE Transactions on Information Forensics and Security*, 16, 2941–2956.
- Hespanha, J.P., Naghshtabrizi, P. & Xu, Y. (2007) A survey of recent results in networked control systems. *Proceedings of the IEEE*, 95(1), 138–162.
- Jadhav, A. & Mutreja, S. (2020) *Autonomous ships market*. Allied Market Research-Freight & Logistics, Portland, Oregon, USA.
- Jallad, K.A., Aljndi, M., & Desouki, M.S. (2020) Anomaly detection optimization using big data and deep learning to reduce false-positive. *Journal of Big Data*, 7(68).
- Kang, T.U., Song, H.M., Jeong, S. & Kim, H.K. (2018) Automated reverse engineering and attack for CAN using OBD-II. In: 2018 IEEE 88<sup>th</sup>



- vehicular technology conference (VTC-Fall), pp. 1–7. Chicago, IL, USA: IEEE.
- Kavallieratos, G., Katsikas, S. & Gkioulos, V. (2019) Cyber-attacks against the autonomous ship. In: Katsikas, S.K., Cuppens, F., Cuppens, N., Lambrinouidakis, C., Antón, A., Gritzalis, S., Mylopoulos, J. & Kalloniatis, C. (Eds.) *Computer security*. Cham: Springer International Publishing, pp. 20–36.
- Kridner, J., Coley, G. & Day, R.P. (2021) *Beaglebone black system reference manual*. Available at: <https://github.com/beagleboard/beaglebone-black/wiki/System-Reference-Manual> [Accessed: 2021-10-26].
- Lund, M.S., Hareide, O.S. & Jøsok, Ø. (2018) An attack on an integrated navigation system. *Necesse*, 3, 149–163.
- Mun, H., Han, K. & Lee, D.H. (2020) Ensuring safety and security in CAN-based automotive embedded systems: A combination of design optimization and secure communication. *IEEE Transactions on Vehicular Technology*, 69(7), 7078–7091.
- NTNU. (2021) *Autonomous all-electric passenger ferries for urban water transport (autoferry)*. Available at: <https://www.ntnu.edu/autoferry> [Accessed: 2021-09-29].
- Nvidia. (2021) *Jetson AGX Xavier developer kit*. Available at: <https://developer.nvidia.com/embedded/jetson-agx-xavier-developer-kit> [Accessed: 2021-10-26].
- Pinto, J., Dias, P.S., Martins, R., Fortuna, J., Marques, E. & Sousa, J. (2013) The LSTS toolchain for networked vehicle systems. In: 2013 MTS/IEEE OCEANS—Bergen, pp. 1–9. Bergen, Norway: IEEE.
- Quinton, L. (2021). *Wärtsilä to develop autonomous, zero emission barge for port of Rotterdam*. Helsinki, Finland: Wärtsilä Corporation Press Release.
- Research and Markets. (2021) *Global autonomous ships market report 2021–2030: Increasing threat of cybersecurity and privacy is expected to limit market growth*. Dublin, Ireland: GlobeNewswire.
- Rolls-Royce. (2018) *Rolls-Royce and finferries demonstrate world's first fully autonomous ferry*. Rolls-Royce Press Release.
- SBG Systems. (2021) *Ellipse series*. Available at: [https://www.sbg-systems.com/products/ellipse-series/#ellipse-d\\_rtk\\_gnss\\_ins](https://www.sbg-systems.com/products/ellipse-series/#ellipse-d_rtk_gnss_ins) [Accessed: 2021-10-22].
- Senti Systems. (2021) *SentiPack*. Available at: <https://sentisolution.com/wp-content/uploads/2020/08/datasheet.pdf> [Accessed: 2021-10-26].
- Silverajan, B., Ocaik, M. & Nagel, B. (2018) Cybersecurity attacks and defences for unmanned smart ships. In: 2018 IEEE international conference on internet of things (iThings) and IEEE green computing and communications (GreenCom) and IEEE cyber, physical and social computing (CPSCom) and IEEE smart data (SmartData), pp. 15–20. Halifax, NS, Canada: IEEE.
- Solnør, P. (2020) A Cryptographic toolbox for feedback control systems. *Modeling, Identification and Control*, 41(4), 313–332.
- Stinson, D.R. & Paterson, M. (2018) *Cryptography: Theory and practice*, 4<sup>th</sup> edition. Boca Raton: Chapman and Hall/CRC.
- Sun, X., Yu, F.R. & Zhang, P. (2021) A survey on cyber-security of connected and autonomous vehicles (CAVs). *IEEE Transactions on Intelligent Transportation Systems*, 1–20.
- Tan, Y., Wang, J., Liu, J. & Zhang, Y. (2020) Unmanned systems security: Models, challenges, and future directions. *IEEE Network*, 34(4), 291–297.
- Teixeira, A., Pérez, D., Sandberg, H., & Johansson, K.H. (2012) Attack models and scenarios for networked control systems. In: *Proceedings of the 1st international conference on high confidence networked systems HiCoNS'12, Beijing, China*. New York, NY: Association for Computing Machinery, pp. 55–64.
- Teixeira, R.R., Maurell, I.P. & Drews, P.L.J. (2020) Security on ROS: analyzing and exploiting vulnerabilities of ROS-based systems. In: 2020 Latin American robotics symposium (LARS), 2020 Brazilian symposium on robotics (SBR) and 2020 workshop on robotics in education (WRE), pp. 1–6. Natal, Brazil: IEEE.
- Tuohy, S., Glavin, M., Hughes, C., Jones, E., Trivedi, M. & Kilmartin, L. (2015) Intra-vehicle networks: A review. *IEEE Transactions on Intelligent Transportation Systems*, 16(2), 534–545.
- U-blox. (2021) *ZED-F9P module*. Available at: <https://www.u-blox.com/en/product/zed-f9p-module> [Accessed: 2021-10-22].
- Vinnem, J.E. & Utne, I.B. (2018) Risk from cyberattacks on autonomous ships. In: Haugen, S., Barros, A., van Gulijk, C., Kongsvik, T. & Vinnem, J.E. (Eds.) *Safety and reliability—safe societies in a changing world: Proceedings of the ESREL 2018 June 17–21, 2018, Trondheim, Norway*, 1st edition. London: CRC Press, pp. 1485–1492.
- Volden, Ø., Solnør, P., Petrovic, S. & Fossen, T.I. (2021). Secure and efficient transmission of vision-based feedback control signals. *Journal of Intelligent & Robotic Systems*, 103(2), 26.
- Wang, Q. & Yang, H. (2019) A survey on the recent development of securing the networked control systems. *Systems Science & Control Engineering*, 7(1), 54–64.
- Wollschlaeger, M., Sauter, T. & Jasperneite, J. (2017) The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0. *IEEE Industrial Electronics Magazine*, 11(1), 17–27.
- Wu, H. & Preneel, B. (2014) AEGIS: A fast authenticated encryption algorithm. In: Lange, T., Lauter, K. & Lisoněk, P. (Eds.) *Selected areas in cryptography—SAC 2013*. Berlin, Heidelberg: Springer, pp. 185–201.
- Wu, W., Li, R., Xie, G., An, J., Bai, Y., Zhou, J. & Li, K. (2020) A survey of intrusion detection for in-vehicle networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(3), 919–933.
- Yağdereli, E., Gemci, C. & Aktaş, A. Z. (2015) A study on cyber-security of autonomous and unmanned vehicles. *The Journal of Defense Modeling and Simulation*, 12(4), 369–381.
- Zhang, X., Han, Q., Ge, X., Ding, D., Ding, L., Yue, D. & Peng, C. (2020) Networked control systems: A survey of trends and techniques. *IEEE/CAA Journal of Automatica Sinica*, 7(1), 1–17.

**How to cite this article:** Solnør, P., Volden, Ø., Gryte, K., Petrovic, S. & Fossen, T.I. (2022) Hijacking of unmanned surface vehicles: A demonstration of attacks and countermeasures in the field. *Journal of Field Robotics*, 1–19. <https://doi.org/10.1002/rob.22068>