Mechanical Systems and Signal Processing 222 (2025) 111778

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

journal homepage: www.elsevier.com/locate/ymssp

A unified sensor and actuator fault diagnosis in digital twins for remote operations

Agus Hasan [a,*], Pierluigi Salvo Rossi [b]

[a] *Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Larsgårdsvegen 2, Ålesund, 6009, Norway*
[b] *Department of Electronic Systems, Norwegian University of Science and Technology, O. S. Bragstads plass 2, Trondheim, 7034, Norway*

ARTICLE INFO

Communicated by Y. Lei

Keywords:
Fault diagnosis
Digital twins
Kalman filter

ABSTRACT

This paper explores the development of a unified hybrid approach for sensor and actuator fault diagnosis in digital twins for remote operations. Central to this approach is the implementation of a robust adaptive Kalman filter algorithm, which forms the backbone of the proposed unified algorithm. The essence of this unified algorithm lies in its capability to effectively filter the sensor measurements. The algorithm is enriched with tuning parameters, offering flexibility in adjusting the convergence rate to suit operational requirements. Noteworthy for its robustness, our approach excels in handling uncertainties and diverse types of faults, including drift, bias, noise, and freeze fault scenarios. Through comprehensive simulation and experimental evaluations conducted on a small surface vessel, the method demonstrates remarkable proficiency in accurately identifying sensor and actuator faults. This precision enables early detection and prompt mitigation of anomalies, contributing to heightened operational resilience.

1. Introduction

In remote operations, challenges like limited physical access and data latency necessitate data-driven fault diagnosis. Ensuring timely responses is crucial, demanding strategic measures to mitigate delays [1,2]. Digital twin technology, a virtual representation of physical objects or systems, is crucial for remote operations, offering continuous monitoring, analysis, and predictive maintenance capabilities. It allows operators to make informed decisions based on real-time data, preventing unplanned downtime [3–7]. In remote operations, digital twins rely on integrated sensor and actuator systems. Sensors gather data points such as temperature, pressure, and vibration, providing real-time insights, while actuators enable remote control and intervention, facilitating adjustments based on sensor data [8]. Both sensors and actuators are susceptible to malfunction and failure, which can have significant implications to operational safety. Over time, sensors may degrade due to wear and tear. This can result from exposure to harsh environmental conditions, mechanical stress, or chemical corrosion, ultimately leading to reduced accuracy or complete failure [9]. Moreover, many sensors rely on electronic components like transistors and microchips. These components can fail due to manufacturing defects or electrical issues. Meanwhile, actuators often involve moving parts, which are prone to mechanical wear. Friction, stress, and fatigue can lead to breakdowns in the mechanical components of an actuator. Actuators are controlled electronically, and failures can occur in the control systems. This may be due to electrical faults, software bugs, or communication issues between the control unit and the actuator [10].

Fig. 1 provides an illustrative schematic diagram representing the concept of digital twins in the context of remote operations for surface vessels. This digital twin model is integral to ensuring the seamless operation and maintenance of such vessels. The

* Corresponding author.
E-mail address: agus.hasan@ntnu.no (A. Hasan).

https://doi.org/10.1016/j.ymssp.2024.111778
Received 13 December 2023; Received in revised form 7 June 2024; Accepted 24 July 2024
Available online 31 July 2024

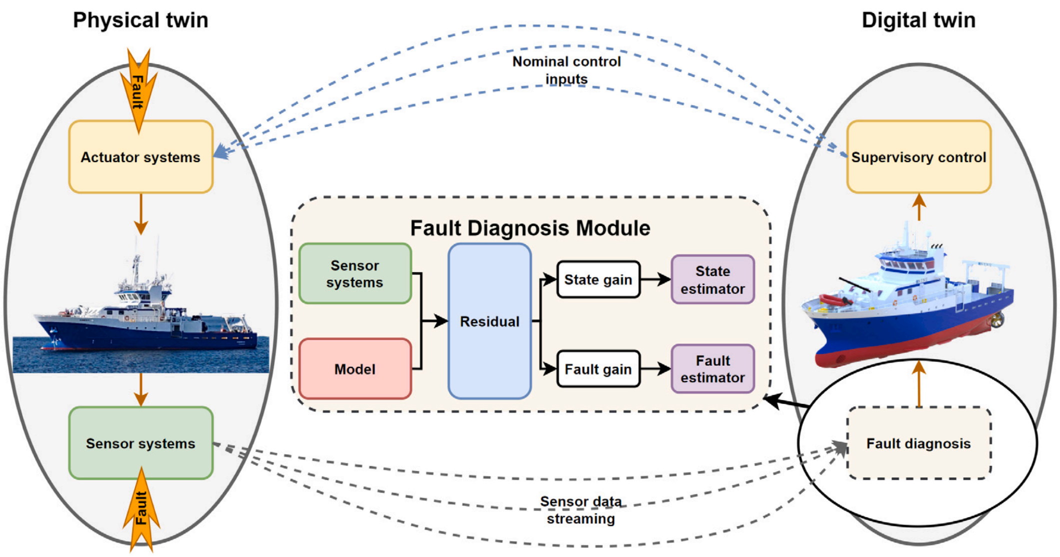

Fig. 1. A schematic diagram of digital twins for remote operation of a surface vessel completed with a fault diagnosis module.

system starts with the acquisition of data from various sensors deployed on the vessels. These sensors are responsible for capturing a wide range of information, including environmental conditions, ship performance parameters, and safety-related data. The data collected by the sensor systems is transmitted to a central component known as the fault diagnosis module. This module serves as an intelligent system that processes the incoming sensor data in real-time. The primary role of the fault diagnosis module is to evaluate the state of health of both sensor and actuator systems. It monitors sensor performance and detects any anomalies or deviations from expected data patterns. If the fault diagnosis module detects actuator faults, it sends a signal indicating that the actuators are not functioning correctly. Similarly, if sensor faults are detected, the signal received by the digital twin is compromised, as it is based on incorrect or unreliable data.

1.1. Literature review

Diagnosing faults can be approached through data-driven, model-based, or hybrid approaches [11,12]. Data-driven fault diagnosis hinges on the utilization of extensive datasets obtained from sensors and monitoring devices embedded in the physical asset [13,14]. Machine learning algorithms, such as neural networks and support vector machines, play a pivotal role in analyzing these datasets [15,16]. Their objective is to discern patterns within the data that may serve as indicators of potential faults. The richness of the dataset allows for the automated identification of anomalies, enabling timely intervention and fault mitigation. The challenge in data-driven fault diagnosis is twofold [17]. Firstly, it involves the effective management of large volumes of data generated by numerous sensors. This includes the task of ensuring data quality, dealing with missing or noisy data, and handling the sheer quantity of information. Secondly, the interpretation of these datasets becomes crucial for effective fault diagnosis. Understanding the nuances within the data patterns and discerning genuine faults from normal variations are intricate tasks. As the datasets grow in complexity, striking the right balance between model complexity and interpretability becomes paramount [18].

Model-based fault diagnosis is a method that revolves around constructing a mathematical representation of the physical system within the digital twin [19]. This representation, often in the form of mathematical equations or models, shows the expected behavior of the system under normal operating conditions. The core principle behind this approach is to compare the predicted behavior derived from the model with the actual observed behavior of the physical asset. Discrepancies or deviations between these two sets of behaviors serve as indicators of potential faults or anomalies within the system [20]. One of the key advantages of model-based fault diagnosis lies in its ability to provide detailed insights into the dynamics of the system [21]. When a comprehensive understanding of the underlying physics and mechanics is available, this approach can offer precise and accurate fault detection. By leveraging the knowledge encoded in the mathematical model, it becomes possible to anticipate and identify deviations from expected behaviors, facilitating early intervention and preventive measures. However, the effectiveness of model-based fault diagnosis is contingent on the accuracy of the mathematical model [22]. Constructing an accurate model requires a thorough understanding of the system's dynamics, components, and interactions. Any inaccuracies or uncertainties in the model can lead to false positives or negatives in fault diagnosis. Moreover, maintaining and updating the model to reflect changes in the physical system over time poses a challenge [23].

Hybrid fault diagnosis represents a sophisticated approach that seamlessly integrates aspects of both data-driven and model-based methodologies [24]. This hybridization is designed to capitalize on the strengths of each approach, creating a more robust fault diagnosis system. The fundamental idea is to leverage the flexibility and adaptability of data-driven techniques while incorporating the structured insights provided by model-based methods. This combination aims to enhance fault diagnosis performance, especially in scenarios where either developing comprehensive models is challenging or the available data is inherently noisy and complex. In a hybrid fault diagnosis system, data-driven techniques, such as machine learning algorithms, are employed to analyze vast datasets collected from sensors and monitoring devices. These techniques excel in identifying patterns and anomalies within the data, facilitating the detection of potential faults. Simultaneously, a mathematical model of the physical system is integrated into the digital twin. This model provides a structured understanding of the system's dynamics and expected behaviors. Deviations between the predicted behavior from the model and the observed behavior from the data serve as crucial indicators of potential faults. The synergy achieved through hybrid fault diagnosis is particularly advantageous when faced with real-world complexities [25]. Developing accurate and comprehensive models for highly dynamic or complex systems can be a formidable task, and data-driven techniques provide a practical alternative by learning from observed behaviors. On the other hand, data-driven approaches may struggle when faced with limited or noisy data, and the structured insights offered by model-based methods become invaluable in such cases. By combining these approaches, hybrid fault diagnosis not only enhances the accuracy of fault detection but also improves the overall reliability of the system [26].

1.2. Contribution of this paper

This paper introduces a pioneering unified architecture designed for the diagnosis of sensor and actuator faults in digital twins utilized for remote operations. The paper's innovation lies in the transformation of sensor faults into the state space model, enabling the concurrent diagnosis of sensor and actuator faults. Notably, the proposed adaptive extended Kalman filter deviates from traditional approaches by directly calculating fault parameters from measurements, rather than considering them as augmented state variables. This integrated methodology facilitates early anomaly detection and management, streamlining decision-making processes within digital twins. This contribution establishes a foundation for predictive maintenance strategies, encourages further research, and sets a new standard for optimizing the efficiency and dependability of complex systems in remote operational scenarios.

1.3. Organization of this paper

This paper provides a comprehensive exploration of sensor and actuator fault diagnosis within the context of digital twins for remote operations. The systematic structure unfolds as follows: Section 2 formulates the problem statement, elucidating the associated challenges. Moving forward, Section 3 is subdivided into four key sub-sections: 3.1 presents an actuator fault diagnosis algorithm, 3.2 focuses on a sensor fault diagnosis algorithm, 3.3 introduces a unified sensor and actuator fault diagnosis algorithm, and 3.4 extends the discussion to address the complexities of nonlinear systems. In Section 4, we present system modeling, numerical simulations, conduct a comparative analysis with alternative methods, and substantiate real-world applicability through illustrative use cases. Section 5 presents our findings, offering a synthesis of the research outcomes and articulating future avenues for development.

2. Problem formulation

The behavior model of a physical asset provides a mathematical representation of how the asset behaves over time. In this paper, this behavior model is expressed in a state space form, which is a common framework for modeling dynamic systems. The state space form consists of two key equations:

$$\boldsymbol{x}(k) = \boldsymbol{A}(k)\boldsymbol{x}(k-1) + \boldsymbol{B}(k)\boldsymbol{u}(k) + \boldsymbol{w}(k) \tag{1}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}(k)\boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{2}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^n$ represents the state of the physical asset at time k, $\boldsymbol{A}(k) \in \mathbb{R}^{n\times n}$ is a matrix that describes how the state evolves over time, $\boldsymbol{B}(k) \in \mathbb{R}^{n\times p}$ represents the influence of control inputs $\boldsymbol{u}(k) \in \mathbb{R}^p$ on the state, $\boldsymbol{w}(k) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}(k)) \in \mathbb{R}^n$ denotes any process noise or disturbances affecting the state, $\boldsymbol{y}(k) \in \mathbb{R}^m$ represents the output or measurements of the physical asset at time k, $\boldsymbol{C}(k) \in \mathbb{R}^{m\times n}$ is a matrix that maps the state to the observed outputs, and $\boldsymbol{v}(k) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}(k)) \in \mathbb{R}^m$ represents measurement noise or errors. The first equation essentially describes how the internal state of the asset changes with time, considering both its own dynamics and input dynamics, while the second equation describes how the state of the asset is related to the measurements that can be obtained from it.

In the context of system modeling and control, faults in the actuator and sensor systems are often represented as parameters that are either constant or piece-wise constant, and these parameters impact the dynamic model (1) and the measurement Eq. (2). If we denote these fault parameters as $\boldsymbol{\theta}_a \in \mathbb{R}^p$ for actuator faults and $\boldsymbol{\theta}_s \in \mathbb{R}^m$ for sensor faults, the state space model can be modified to include the effects of these faults. The modified state space model is as follows:

$$\boldsymbol{x}(k) = \boldsymbol{A}(k)\boldsymbol{x}(k-1) + \boldsymbol{B}(k)\boldsymbol{u}(k) + \boldsymbol{\Phi}(k)\boldsymbol{\theta}_a + \boldsymbol{w}(k) \tag{3}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}(k)\boldsymbol{x}(k) + \boldsymbol{\Psi}(k)\boldsymbol{\theta}_s + \boldsymbol{v}(k) \tag{4}$$

where $\boldsymbol{\Phi}(k) \in \mathbb{R}^{n\times p}$ captures the impact of actuator faults, and $\boldsymbol{\Psi}(k) \in \mathbb{R}^{m\times m}$ models the impact of sensor faults at time k. The primary objective of our research, as outlined in this paper, is to propose a methodology that can effectively identify and quantify the extent of faults represented by the parameters $\boldsymbol{\theta}_a$ (associated with actuator faults) and $\boldsymbol{\theta}_s$ (related to sensor faults).

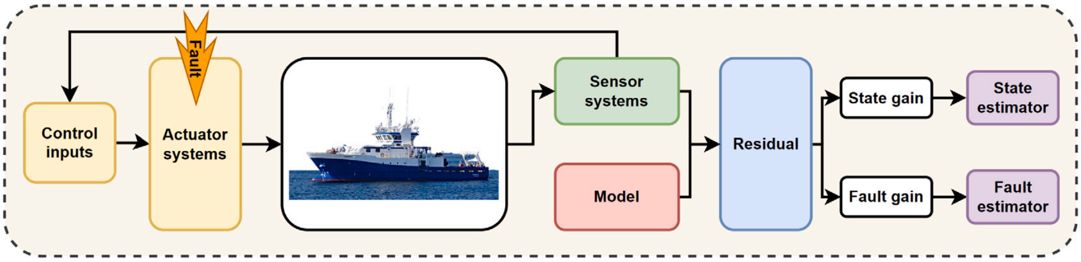

Fig. 2. A schematic representation illustrating the introduction of actuator faults into the physical assets.

3. Proposed methodology

This section is structured into four sub-sections, each focusing on distinct aspects of fault diagnosis. In Sub- Section 3.1, we introduce a comprehensive methodology tailored for actuator fault diagnosis. Sub- Section 3.2 provides an in-depth exploration of a dedicated methodology designed for sensor fault diagnosis. Delving further into fault detection strategies, Sub- Section 3.3 presents a methodology specifically crafted for addressing joint sensor and actuator faults. Expanding the scope to accommodate nonlinear systems, Sub- Section 3.4 explores an extension of the methodology to effectively handle the complexities associated with nonlinear systems. Each sub-section contributes to a holistic understanding of fault diagnosis techniques, offering tailored approaches for various facets of system monitoring and ensuring a comprehensive coverage of fault scenarios.

3.1. Actuator fault diagnosis

In this section, we dedicate our efforts to the formulation and implementation of an algorithm designed for the purpose of actuator fault diagnosis. As visualized in Fig. 2, actuator faults are not uncommon occurrences, and they can be attributed to a multitude of underlying factors. These factors encompass a wide spectrum, including but not limited to material fatigue, excessive torque, and power supply deficiencies. It is important to recognize that the severity of these actuator faults can vary significantly, ranging from minor operational hiccups to potentially hazardous situations that necessitate the immediate cessation of the operation in question. Therefore, our objective here is to provide a robust algorithm that can effectively identify, classify, and address actuator faults of varying degrees of severity, enhancing the safety and reliability of the systems in which these actuators are employed. Subsequently, we will explore the details of our algorithm, explaining its workings and demonstrating its utility in addressing actuator faults and ensuring operational integrity.

Let us examine the state space model or digital twin of a system that incorporates actuator faults. The primary aim is to derive an estimate for the unidentified actuator fault represented by $\boldsymbol{\theta}_a$, leveraging the measurements captured by $\boldsymbol{y}(k)$. The state space model under actuator fault is given as follows:

$$\boldsymbol{x}(k) = \boldsymbol{A}(k)\boldsymbol{x}(k-1) + \boldsymbol{B}(k)\boldsymbol{u}(k) + \boldsymbol{\Phi}(k)\boldsymbol{\theta}_a + \boldsymbol{w}(k) \tag{5}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}(k)\boldsymbol{x}(k) + \boldsymbol{v}(k) \tag{6}$$

The estimation algorithm employed in this paper relies on a modified adaptive Kalman filter. A key distinction between our proposed algorithm and the traditional Kalman filter lies in our approach to directly estimating the fault parameter from the measurement. In the traditional Kalman filter, first we define:

$$\boldsymbol{\zeta}(k) = \begin{pmatrix} \boldsymbol{x}(k) \\ \boldsymbol{\theta}_a(k) \end{pmatrix} \tag{7}$$

with $\boldsymbol{\theta}_a(k) = \boldsymbol{\theta}_a(k-1) = \boldsymbol{\theta}_a$. Thus, (5)–(6) can be written as:

$$\boldsymbol{\zeta}(k) = \mathscr{A}(k)\boldsymbol{\zeta}(k-1) + \mathscr{B}(k)\boldsymbol{u}(k) + \mathscr{W}(k) \tag{8}$$

$$\boldsymbol{y}(k) = \mathscr{C}(k)\boldsymbol{\zeta}(k) + \boldsymbol{v}(k) \tag{9}$$

where:

$$\mathscr{A}(k) = \begin{pmatrix} \boldsymbol{A}(k) & \boldsymbol{\Phi}(k) \\ \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \tag{10}$$

$$\mathscr{B}(k) = \begin{pmatrix} \boldsymbol{B}(k) \\ \boldsymbol{0} \end{pmatrix} \tag{11}$$

$$\mathscr{C}(k) = \begin{pmatrix} \boldsymbol{C}(k) & \boldsymbol{0} \end{pmatrix} \tag{12}$$

$$\mathscr{W}(k) = \begin{pmatrix} \boldsymbol{w}(k) \\ \boldsymbol{0} \end{pmatrix} \tag{13}$$

The next step is to implement the Kalman filter for the transformed system (8)–(9). While this approach may work well for systems with linear behavior, it can produce divergent results for nonlinear systems. Our algorithm improves the process by directly inferring the fault parameter from the measured data. This modification increases the efficiency of the estimation process and enhances the algorithm's effectiveness in accurately identifying and addressing faults within the system. Direct estimation from measurements not only simplifies the computational steps but also improves the algorithm's adaptability to real-time applications, making it a valuable asset in scenarios where timely fault detection and correction are critical. In this case, the fault parameter and state estimators are given as follow:

$$\hat{\theta}_a(k) = \hat{\theta}_a(k-1) + \boldsymbol{\Theta}(k)\tilde{\boldsymbol{y}}(k) \tag{14}$$

$$\hat{\boldsymbol{x}}(k|k) = \boldsymbol{A}(k)\hat{\boldsymbol{x}}(k-1|k-1) + \boldsymbol{B}(k)\boldsymbol{u}(k) + \boldsymbol{\Phi}(k)\hat{\theta}_a(k-1) + \boldsymbol{K}(k)\tilde{\boldsymbol{y}}(k) + \boldsymbol{\Pi}(k)\left[\hat{\theta}_a(k) - \hat{\theta}_a(k-1)\right] \tag{15}$$

Examining (14), we can discern that the estimation of the actuator fault at time k involves the incorporation of the measurement error $\tilde{\boldsymbol{y}}(k)$ into the previous estimation. It is noteworthy that the determination of the parameter estimation gain $\boldsymbol{\Theta}(k)$ is a subsequent step in this process. Moving on to the state estimation represented by (15), a similar methodology is employed as in parameter estimation. However, a distinctive aspect is the involvement of the Kalman gain $\boldsymbol{K}(k)$ in the output injection step. In essence, the estimation of both actuator fault and system state follows a recursive process, wherein the latest measurement is assimilated into the previous estimation. The adjustment of parameters, whether it be the parameter estimation gain $\boldsymbol{\Theta}(k)$ or the Kalman gain $\boldsymbol{K}(k)$, is integral to refining the accuracy of the estimations. This recursive nature of the algorithm contributes to its adaptability and effectiveness in tracking and responding to dynamic changes in the system over time. The Kalman gain $\boldsymbol{K}(k)$ can be calculated using the standard Kalman filter formula as follows [27]:

$$\boldsymbol{P}(k|k-1) = \boldsymbol{A}(k)\boldsymbol{P}(k-1|k-1)\boldsymbol{A}(k)^{\intercal} + \boldsymbol{Q}(k) \tag{16}$$

$$\boldsymbol{\Sigma}(k) = \boldsymbol{C}(k)\boldsymbol{P}(k|k-1)\boldsymbol{C}(k)^{\intercal} + \boldsymbol{R}(k) \tag{17}$$

$$\boldsymbol{K}(k) = \boldsymbol{P}(k|k-1)\boldsymbol{C}(k)^{\intercal}\boldsymbol{\Sigma}(k)^{-1} \tag{18}$$

$$\boldsymbol{P}(k|k) = \left[\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{C}(k)\right]\boldsymbol{P}(k|k-1) \tag{19}$$

In the given context, $\boldsymbol{P}(k|k) \in \mathbb{R}^{n\times n}$ represents the posteriori estimate covariance matrix. The determination of the parameter estimation gain $\boldsymbol{\Theta}(k)$ and the state estimation gain $\boldsymbol{\Pi}(k)$ in (15) is derived from the following formula [28]:

$$\boldsymbol{\Pi}(k) = \left[\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{C}(k)\right]\boldsymbol{A}(k)\boldsymbol{\Pi}(k-1) + \left[\boldsymbol{I} - \boldsymbol{K}(k)\boldsymbol{C}(k)\right]\boldsymbol{\Phi}(k) \tag{20}$$

$$\boldsymbol{\Omega}(k) = \boldsymbol{C}(k)\boldsymbol{A}(k)\boldsymbol{\Pi}(k-1) + \boldsymbol{C}(k)\boldsymbol{\Phi}(k) \tag{21}$$

$$\boldsymbol{\Lambda}(k) = \left[\lambda\boldsymbol{\Sigma}(k) + \boldsymbol{\Omega}(k)\boldsymbol{S}(k-1)\boldsymbol{\Omega}(k)^{\intercal}\right]^{-1} \tag{22}$$

$$\boldsymbol{\Theta}(k) = \boldsymbol{S}(k-1)\boldsymbol{\Omega}(k)^{\intercal}\boldsymbol{\Lambda}(k) \tag{23}$$

$$\boldsymbol{S}(k) = \frac{1}{\lambda}\boldsymbol{S}(k-1) - \frac{1}{\lambda}\boldsymbol{S}(k-1)\boldsymbol{\Omega}(k)^{\intercal}\boldsymbol{\Lambda}(k)\boldsymbol{\Omega}(k)\boldsymbol{S}(k-1) \tag{24}$$

This formula shows the computation of both the parameter and state estimation gains, acting as a pivotal step in refining the accuracy of the overall estimation process. The parameter estimation gain, $\boldsymbol{\Theta}(k)$, influences how the system adjusts its understanding of the fault parameter based on the available measurements, while the state estimation gain, $\boldsymbol{\Pi}(k)$, plays a crucial role in incorporating the measurement information into the overall state estimation. The nuances of these gains, determined through this formula, significantly contribute to the adaptability and precision of the estimation algorithm in capturing the dynamic behavior of the system over time. It is important to note that the algorithm (20)–(24) incorporates three auxiliary variables, namely $\boldsymbol{\Omega}(k)$, $\boldsymbol{\Lambda}(k)$, and $\boldsymbol{S}(k)$, along with a tuning parameter denoted as λ. Initializing the algorithm requires defining the initial values for these variables: $\boldsymbol{P}(0|0)$, $\boldsymbol{\Pi}(0)$, $\boldsymbol{S}(0)$, $\hat{\theta}_a(0)$, and $\hat{\boldsymbol{x}}(0|0)$. The initial value $\boldsymbol{P}(0|0)$ signifies the initial posteriori estimate covariance matrix, setting the foundation for subsequent estimations. $\boldsymbol{\Pi}(0)$ is the initial state estimation gain, influencing the assimilation of measurement data into the system state estimate. $\boldsymbol{S}(0)$ represents the initial state covariance matrix, capturing the uncertainty associated with the initial state estimate. Initializing $\hat{\theta}_a(0)$ and $\hat{\boldsymbol{x}}(0|0)$ involves specifying the initial estimates for the actuator fault parameter and the system state, respectively. Effectively setting these initial values is critical for the algorithm's convergence and accurate performance.

The tuning parameter λ is commonly referred to as the forgetting factor. The choice of λ plays a significant role in shaping the algorithm's behavior. When λ is smaller, the algorithm exhibits a faster transient response, making it more responsive to short-term variations but also more sensitive to noise. On the other hand, a larger λ yields a slower transient response, resulting in smoother estimates over time. The selection of λ thus represents a trade-off between responsiveness to dynamic changes and robustness against noise, allowing the algorithm's behavior to be tailored to the specific characteristics and requirements of the system under consideration.

3.2. Sensor fault diagnosis

In this section, we formulate an algorithm specifically designed for sensor fault diagnosis. Given the critical impact of sensor faults on the outcome of autonomous operations, detecting and estimating these faults becomes paramount. Upon the occurrence of a sensor fault, our algorithm is designed to estimate its magnitude, as illustrated in Fig. 3. The underlying concept of the algorithm mirrors that of actuator fault diagnosis. The key lies in transforming the sensor fault into the state space equation, accomplished through the utilization of a sensor filter. This approach allows us to seamlessly integrate the detection and estimation of sensor faults

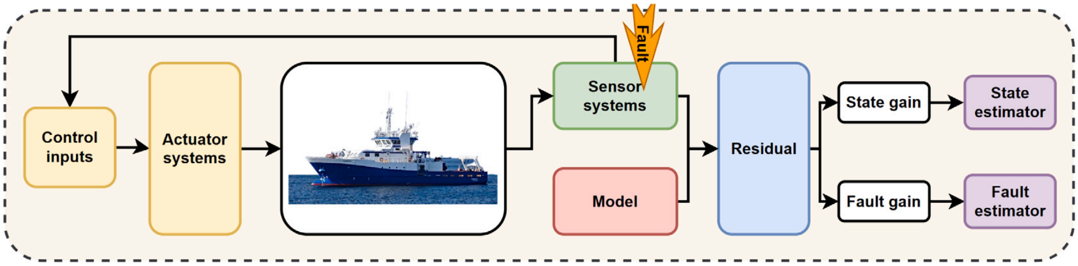

Fig. 3. A schematic representation illustrating the introduction of sensor faults into the physical assets.

into the broader context of the system's state dynamics, facilitating a comprehensive approach to fault diagnosis in autonomous operations.

To this end, let us consider the following sensor fault problem:

$$\boldsymbol{x}(k) = \boldsymbol{A}(k)\boldsymbol{x}(k-1) + \boldsymbol{B}(k)\boldsymbol{u}(k) + \boldsymbol{w}(k) \tag{25}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}(k)\boldsymbol{x}(k) + \boldsymbol{\Psi}(k)\boldsymbol{\theta}_s + \boldsymbol{v}(k) \tag{26}$$

The sensor fault, denoted as $\boldsymbol{\theta}_s$, is introduced into the measurement equation, thereby influencing the behavior of the system. To estimate this fault, the initial step involves the application of a filtering process to the sensor measurements, as expressed by the following equation:

$$\boldsymbol{z}(k) = \left(\boldsymbol{I} - \boldsymbol{A}_f\Delta t\right)\boldsymbol{z}(k-1) + \boldsymbol{A}_f\Delta t\boldsymbol{C}(k)\boldsymbol{x}(k-1) + \boldsymbol{A}_f\Delta t\boldsymbol{\Psi}(k)\boldsymbol{\theta}_s + \boldsymbol{A}_f\Delta t\boldsymbol{v}(k) \tag{27}$$

Here $\boldsymbol{z}(k) \in \mathbb{R}^m$ is the filtered signals, $\boldsymbol{A}_f \in \mathbb{R}^{m\times m}$ is a positive definite matrix, and Δt is the sampling time. Augmenting (25) and (27), we obtain:

$$\boldsymbol{x}(k) = \boldsymbol{A}(k)\boldsymbol{x}(k-1) + \boldsymbol{B}(k)\boldsymbol{u}(k) + \boldsymbol{w}(k) \tag{28}$$

$$\boldsymbol{z}(k) = \boldsymbol{A}_f\Delta t\boldsymbol{C}(k)\boldsymbol{x}(k-1) + \left(\boldsymbol{I} - \boldsymbol{A}_f\Delta t\right)\boldsymbol{z}(k-1) + \boldsymbol{A}_f\Delta t\boldsymbol{\Psi}(k)\boldsymbol{\theta}_s + \boldsymbol{A}_f\Delta t\boldsymbol{v}(k) \tag{29}$$

If we denote:

$$\xi(k) = \begin{pmatrix} \boldsymbol{x}(k) \\ \boldsymbol{z}(k) \end{pmatrix} \in \mathbb{R}^{n+m} \tag{30}$$

Then, we have:

$$\xi(k) = \mathcal{A}(k)\xi(k-1) + \mathcal{B}(k)\boldsymbol{u}(k) + \bar{\boldsymbol{\Psi}}(k)\boldsymbol{\theta}_s + \bar{\boldsymbol{w}}(k) \tag{31}$$

$$\mathcal{Y}(k) = \mathcal{C}(k)\xi(k) \tag{32}$$

where:

$$\mathcal{A}(k) = \begin{pmatrix} \boldsymbol{A}(k) & \boldsymbol{0} \\ \boldsymbol{A}_f\Delta t\boldsymbol{C}(k) & \boldsymbol{I} - \boldsymbol{A}_f\Delta t \end{pmatrix} \in \mathbb{R}^{(n+m)\times(n+m)} \tag{33}$$

$$\mathcal{B}(k) = \begin{pmatrix} \boldsymbol{B}(k) \\ \boldsymbol{0} \end{pmatrix} \in \mathbb{R}^{(n+m)\times p} \tag{34}$$

$$\bar{\boldsymbol{\Psi}}(k) = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{A}_f\Delta t\boldsymbol{\Psi}(k) \end{pmatrix} \in \mathbb{R}^{(n+m)\times m} \tag{35}$$

$$\bar{\boldsymbol{w}}(k) = \begin{pmatrix} \boldsymbol{w}(k) \\ \boldsymbol{A}_f\Delta t\boldsymbol{v}(k) \end{pmatrix} \sim \mathcal{N}(\boldsymbol{0}, \mathcal{Q}(k)) \in \mathbb{R}^{n+m} \tag{36}$$

$$\mathcal{C}(k) = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{I} \end{pmatrix} \in \mathbb{R}^{m\times(n+m)} \tag{37}$$

The resemblance between (31)–(5) and (32)–(6) is evident. This similarity allows us to leverage the algorithm developed for actuator fault diagnosis for the purpose of sensor fault diagnosis. The parallel structure between these equations implies that the methodology and principles employed in diagnosing actuator faults can be effectively applied to address sensor faults. This streamlined approach not only simplifies the implementation of fault diagnosis algorithms but also underscores the interconnected nature of fault detection strategies, emphasizing the potential for a unified diagnostic framework for both sensor and actuator faults within the system.

3.3. Joint sensor and actuator fault diagnosis

In this section, we formulate an algorithm designed to concurrently estimate the magnitudes of both sensor and actuator faults. The coexistence of sensor and actuator faults is a prevalent challenge in systems intended for remote operation, as depicted in

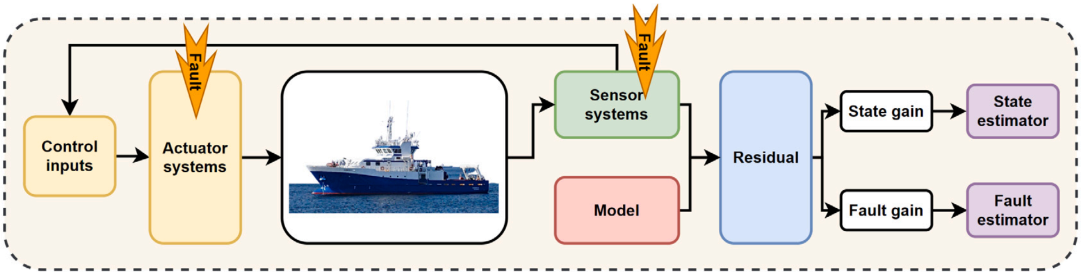

Fig. 4. A schematic representation illustrating the introduction of actuator and sensor faults into the physical assets.

Fig. 4. Addressing these faults simultaneously is crucial for ensuring the reliability and efficiency of remote-operated systems. The proposed algorithm integrates the diagnostic processes for sensor and actuator faults, allowing for a comprehensive approach to fault estimation in scenarios where both types of faults may occur concurrently. This integrated methodology contributes to the overall robustness and adaptability of fault diagnosis in remote operational environments.

We start with the augmented faulty system, as follows:

$$\boldsymbol{x}(k) = \boldsymbol{A}(k)\boldsymbol{x}(k-1) + \boldsymbol{B}(k)\boldsymbol{u}(k) + \boldsymbol{\Phi}(k)\boldsymbol{\theta}_a + \boldsymbol{w}(k) \tag{38}$$

$$\boldsymbol{z}(k) = \boldsymbol{A}_f \Delta t \boldsymbol{C}(k)\boldsymbol{x}(k-1) + \left(\boldsymbol{I} - \boldsymbol{A}_f \Delta t\right)\boldsymbol{z}(k-1) + \boldsymbol{A}_f \Delta t \boldsymbol{\Psi}(k)\boldsymbol{\theta}_s + \boldsymbol{A}_f \Delta t \boldsymbol{v}(k) \tag{39}$$

Applying the identical procedure as in the preceding section, we acquire:

$$\boldsymbol{\xi}(k) = \mathcal{A}(k)\boldsymbol{\xi}(k-1) + \mathcal{B}(k)\boldsymbol{u}(k) + \bar{\boldsymbol{\Psi}}(k)\boldsymbol{\theta} + \bar{\boldsymbol{w}}(k) \tag{40}$$

$$\mathcal{Y}(k) = \mathcal{C}(k)\boldsymbol{\xi}(k) \tag{41}$$

where:

$$\boldsymbol{\theta} = \begin{pmatrix} \boldsymbol{\theta}_a \\ \boldsymbol{\theta}_s \end{pmatrix} \in \mathbb{R}^{p+m} \tag{42}$$

$$\bar{\boldsymbol{\Psi}}(k) = \begin{pmatrix} \boldsymbol{\Phi}(k) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_f \Delta t \boldsymbol{\Psi}(k) \end{pmatrix} \in \mathbb{R}^{(n+m)\times(p+m)} \tag{43}$$

Once more, it is evident that the transformed system mirrors the structure of (5)–(6), enabling the utilization of the same algorithm. The presented formulation demonstrates the capability to concurrently quantify the magnitude of faults in both sensors and actuators. This concurrent fault estimation is a significant and promising outcome for fault diagnosis within the digital twin framework for remote operations. It not only streamlines the diagnostic process but also enhances the comprehensiveness of fault identification, addressing potential challenges arising from both sensor and actuator faults simultaneously.

In summary, the key outcome of this paper is presented in the unified sensor and actuator fault algorithm below:

1. *Step 1*: Initialization: $\boldsymbol{P}(0|0) > 0 \in \mathbb{R}^{(n+m)\times(n+m)}$, $\boldsymbol{\Pi}(0) \in \mathbb{R}^{(n+m)\times(p+m)}$, $\boldsymbol{S}(0) = \omega \boldsymbol{I} \in \mathbb{R}^{p+m}$, $\hat{\boldsymbol{\theta}}(0) \in \mathbb{R}^{p+m}$, and $\hat{\boldsymbol{\xi}}(0|0) \in \mathbb{R}^{n+m}$, $\mathcal{Q}(k) \in \mathbb{R}^{(n+m)\times(n+m)}$, and $\lambda \in \mathbb{R}(0,1)$
2. *Step 2*: Recursions for $k = 1, 2, 3, \ldots$

   (a) Gain calculations

$$\boldsymbol{P}(k|k-1) = \mathcal{A}(k)\boldsymbol{P}(k-1|k-1)\mathcal{A}(k)^\intercal + \mathcal{Q}(k) \tag{44}$$

$$\boldsymbol{\Sigma}(k) = \mathcal{C}(k)\boldsymbol{P}(k|k-1)\mathcal{C}(k)^\intercal \tag{45}$$

$$\boldsymbol{K}(k) = \boldsymbol{P}(k|k-1)\mathcal{C}(k)^\intercal \boldsymbol{\Sigma}(k)^{-1} \tag{46}$$

$$\boldsymbol{P}(k|k) = \left[\boldsymbol{I} - \boldsymbol{K}(k)\mathcal{C}(k)\right]\boldsymbol{P}(k|k-1) \tag{47}$$

$$\boldsymbol{\Pi}(k) = \left[\boldsymbol{I} - \boldsymbol{K}(k)\mathcal{C}(k)\right]\mathcal{A}(k)\boldsymbol{\Pi}(k-1) + \left[\boldsymbol{I} - \boldsymbol{K}(k)\mathcal{C}(k)\right]\bar{\boldsymbol{\Psi}}(k) \tag{48}$$

$$\boldsymbol{\Omega}(k) = \mathcal{C}(k)\mathcal{A}(k)\boldsymbol{\Pi}(k-1) + \mathcal{C}(k)\bar{\boldsymbol{\Psi}}(k) \tag{49}$$

$$\boldsymbol{\Lambda}(k) = \left[\lambda\boldsymbol{\Sigma}(k) + \boldsymbol{\Omega}(k)\boldsymbol{S}(k-1)\boldsymbol{\Omega}(k)^\intercal\right]^{-1} \tag{50}$$

$$\boldsymbol{\Theta}(k) = \boldsymbol{S}(k-1)\boldsymbol{\Omega}(k)^\intercal\boldsymbol{\Lambda}(k) \tag{51}$$

$$\boldsymbol{S}(k) = \frac{1}{\lambda}\boldsymbol{S}(k-1) - \frac{1}{\lambda}\boldsymbol{S}(k-1)\boldsymbol{\Omega}(k)^\intercal\boldsymbol{\Lambda}(k)\boldsymbol{\Omega}(k)\boldsymbol{S}(k-1) \tag{52}$$

   (b) Error calculation

$$\tilde{\mathcal{Y}}(k) = \mathcal{Y}(k) - \mathcal{C}(k)\left[\mathcal{A}(k)\hat{\boldsymbol{\xi}}(k-1|k-1) + \mathcal{B}(k)\boldsymbol{u}(k) + \bar{\boldsymbol{\Psi}}(k)\hat{\boldsymbol{\theta}}(k-1)\right] \tag{53}$$

(c) Fault and state estimation

$$\hat{\theta}(k) = \hat{\theta}(k-1) + \boldsymbol{\Theta}(k)\tilde{\mathcal{Y}}(k) \tag{54}$$

$$\hat{\xi}(k|k) = \mathcal{A}(k)\hat{\xi}(k-1|k-1) + \mathcal{B}(k)\boldsymbol{u}(k) + \bar{\boldsymbol{\Psi}}(k)\hat{\boldsymbol{\theta}}(k-1) + \boldsymbol{K}(k)\tilde{\mathcal{Y}}(k) + \boldsymbol{\Pi}(k)\left[\hat{\boldsymbol{\theta}}(k) - \hat{\boldsymbol{\theta}}(k-1)\right] \tag{55}$$

3.4. Extension to nonlinear systems

In this section, we consider the following nonlinear systems:

$$\xi(k) = \mathcal{A}(k)\xi(k-1) + \boldsymbol{f}(\xi(k-1)) + \mathcal{B}(k)\boldsymbol{u}(k) + \bar{\boldsymbol{\Psi}}(k)\boldsymbol{\theta} + \bar{\boldsymbol{w}}(k) \tag{56}$$

Here, $\boldsymbol{f} \in \mathbb{R}^{n+m}$ represents a continuously differentiable function. It is noteworthy that numerous systems inherently exhibit nonlinearity. Fortunately, for such systems, the proposed method can still be applied provided we can compute the Jacobian matrix of the nonlinear function. In this scenario, linearizing $\boldsymbol{f}$ using the Taylor series yields the Jacobian matrix, expressed as:

$$\mathbf{F}(\bar{\xi}(k-1)) = \left.\frac{\partial \boldsymbol{f}(\xi(k))}{\partial \xi(k)}\right|_{\bar{\xi}(k-1)} \tag{57}$$

Subsequently, this Jacobian matrix will be employed to substitute the state matrix $\mathcal{A}(k)$ in the fault diagnosis gain calculation. This approach is called adaptive Extended Kalman filter.

Similar to the Kalman-filter approach, the proposed method comes with certain assumptions. The first assumption pertains to the bounded nature of matrices $\mathcal{A}(k)$, $\mathcal{B}(k)$, $\mathcal{C}(k)$, and $\bar{\boldsymbol{\Psi}}(k)$. For the majority of physical systems, this assumption is typically satisfied. The second assumption focuses on the uniform observability of the pair $(\mathcal{A}(k), \mathcal{C}(k))$. This assumption ensures a certain level of observability throughout the system. However, the most crucial assumption, applicable to the general parameter estimation problem, is the persistence of excitation condition. This condition is essential for the continuous variation and stimulation of the system, enabling the algorithm to obtain meaningful and accurate parameter estimates over time. The persistence of excitation condition is fundamental for the success of the parameter estimation process. The parameter estimate's convergence prerequisite is the manifestation of persistent excitation in the sequence $\bar{\boldsymbol{\Psi}}(k)$. In this scenario, there are fixed positive constants κ and ξ such that:

$$0 < \kappa \boldsymbol{I} \leq \Sigma_{l=k-\xi}^{k} \bar{\boldsymbol{\Psi}}(l)^{\intercal}\bar{\boldsymbol{\Psi}}(l) \tag{58}$$

If these assumptions are not met, the method may encounter difficulties in accurately estimating the magnitude of faults. Specifically, failure to satisfy these assumptions could impede the effectiveness of the estimation process. However, under conditions where these assumptions are fulfilled, the method exhibits versatility and can be applied to a broad spectrum of systems, provided that the Jacobian matrix can be calculated. The ability to calculate the Jacobian matrix is a key factor in the successful implementation of the method across diverse systems, ensuring its adaptability and effectiveness in fault magnitude estimation.

4. Numerical simulation and experimental validation

In this section, we perform a comprehensive process involving system modeling, numerical simulation, and validation. The latter is carried out utilizing data obtained from experiments conducted with a small unmanned surface vessel (ASV) called Otter, developed by Maritime Robotics. The Otter, designed as a catamaran ASV (see Fig. 5), features electric thrusters fueled by up to four interchangeable battery packs, offering a versatile and powerful propulsion system. Positioned at the rear of the vehicle are two fixed propellers that contribute to its movement. Despite its capacity for diverse maneuvers, the Otter is considered under-actuated due to having fewer actuators than degrees of freedom. To propel the ASV forward, both propellers must generate equal thrust. Conversely, varying thrusts from the propellers allow for left or right maneuvers, showcasing the dynamic control capabilities of the vehicle. The integration of these components allows us to create a robust representation of the system, simulate its behavior under faulty conditions, and validate the model's accuracy by comparing the results with real-world data collected from the Otter vessels during experiments. This approach ensures a thorough and reliable analysis, aligning theoretical modeling with empirical evidence derived from practical applications with the Otter ASV.

4.1. System modeling

We describe the dynamics of the Otter ASV through the following set of equations:

$$\dot{p}(t) = U(t)\cos(\chi(t)) + w_1(t) \tag{59}$$

$$\dot{q}(t) = U(t)\sin(\chi(t)) + w_2(t) \tag{60}$$

$$\dot{U}(t) = a(t) + w_3(t) \tag{61}$$

$$\dot{\chi}(t) = r(t) + w_4(t) \tag{62}$$

These equations show the evolution of key state variables over time. Here, $p(t)$ and $q(t)$ represent the ASV's spatial coordinates, $U(t)$ denotes its velocity, and $\chi(t)$ characterizes its heading angle. The variables $a(t)$ and $r(t)$ denote the acceleration and rate of course angle and are considered as the control inputs, respectively. The noises $w_1(t)$, $w_2(t)$, $w_3(t)$, and $w_4(t)$ are introduced to model

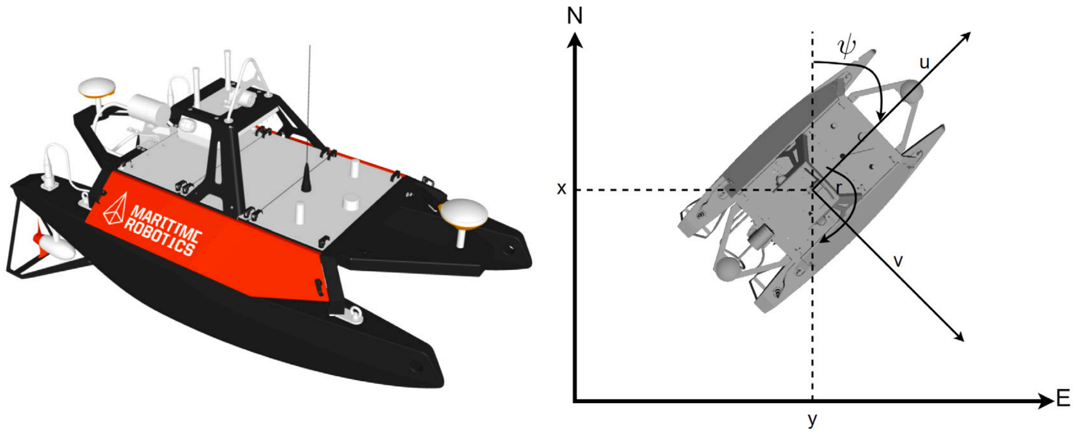

Fig. 5. The Otter (left) and its coordinate modeling.

the inaccuracies. This modeling framework provides a foundation for numerical simulations and further analysis to understand and predict the behavior of the Otter ASV under different conditions. We assume we can measure all variables using the sensor system, such that the measurement matrix is given by:

$$\boldsymbol{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{63}$$

The faults affecting the actuators manifest as discrepancies in the control inputs $a(t)$ and $r(t)$ and are represented by θ_1^a and θ_2^a, respectively. Concurrently, sensor faults correspond to deviations in the measurement sensors responsible for $p(t)$, $q(t)$, $U(t)$, and $\chi(t)$ denoted as θ_1^s, θ_2^s, θ_3^s, and θ_4^s, respectively. Using these notations, the fault vector can be written as:

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_1^a \\ \theta_2^a \\ \theta_1^s \\ \theta_2^s \\ \theta_3^s \\ \theta_4^s \end{pmatrix} \in \mathbb{R}^6 \tag{64}$$

To incorporate these fault considerations, we discretize (59)–(62) using the Euler method. Additionally, by augmenting the measurement filter into the system and without loss of generality for pedagogical purposes taking $\boldsymbol{A}_f = \mu \boldsymbol{I}$ and $\boldsymbol{\Psi} = \nu \boldsymbol{I}$, where $\mu, \nu > 0$, we arrive at the following expression:

$$p(k) = p(k-1) + \Delta t U(k-1)\cos(\chi(k-1)) + \Delta t w_1(k) \tag{65}$$
$$q(k) = q(k-1) + \Delta t U(k-1)\sin(\chi(k-1)) + \Delta t w_2(k) \tag{66}$$
$$U(k) = U(k-1) + \Delta t(1-\theta_1^a)a(k) + \Delta t w_3(k) \tag{67}$$
$$\chi(k) = \chi(k-1) + \Delta t(1-\theta_2^a)r(k) + \Delta t w_4(k) \tag{68}$$
$$z_1(k) = \mu \Delta t p(k-1) + (1-\mu\Delta t)z_1(k-1) + \mu\nu\Delta t\theta_1^s + \mu\Delta t v_1(k) \tag{69}$$
$$z_2(k) = \mu \Delta t q(k-1) + (1-\mu\Delta t)z_2(k-1) + \mu\nu\Delta t\theta_2^s + \mu\Delta t v_2(k) \tag{70}$$
$$z_3(k) = \mu \Delta t U(k-1) + (1-\mu\Delta t)z_3(k-1) + \mu\nu\Delta t\theta_3^s + \mu\Delta t v_3(k) \tag{71}$$
$$z_4(k) = \mu \Delta t \chi(k-1) + (1-\mu\Delta t)z_4(k-1) + \mu\nu\Delta t\theta_4^s + \mu\Delta t v_4(k) \tag{72}$$

The above system resemble (56) with:

$$\mathcal{A}(k) = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \mu\Delta t\boldsymbol{I} & \boldsymbol{I} - \mu\Delta t\boldsymbol{I} \end{pmatrix} \in \mathbb{R}^{8\times 8} \tag{73}$$

$$\boldsymbol{f}(\xi(k-1)) = \begin{pmatrix} \Delta t U(k-1)\cos(\chi(k-1)) \\ \Delta t U(k-1)\sin(\chi(k-1)) \\ \boldsymbol{0} \end{pmatrix} \in \mathbb{R}^8 \tag{74}$$

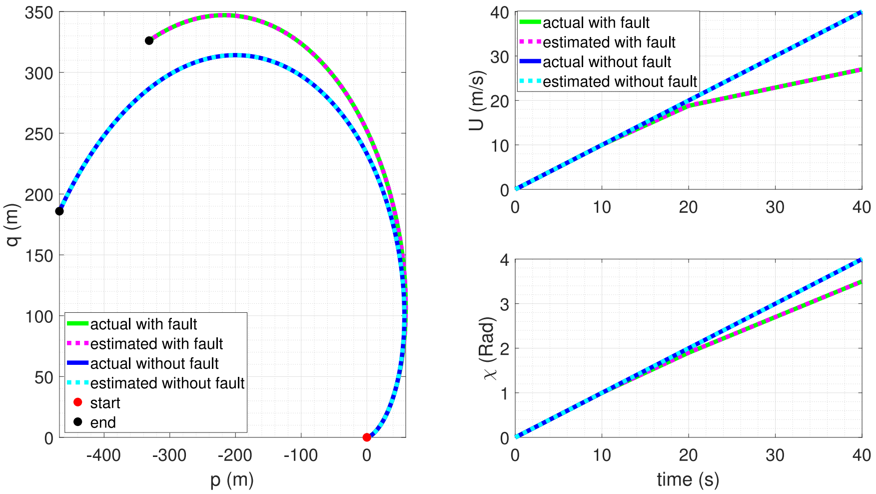

Fig. 6. The evolution of the states under sensor faults.

$$\mathcal{B}(k) = \begin{pmatrix} \boldsymbol{B} \\ \mathbf{0} \end{pmatrix} \in \mathbb{R}^{8\times 2} \tag{75}$$

$$\boldsymbol{B} = \begin{pmatrix} 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & \Delta t \end{pmatrix}^{\top} \in \mathbb{R}^{4\times 2} \tag{76}$$

$$\bar{\boldsymbol{\Psi}}(k) = \begin{pmatrix} -\boldsymbol{B}\mathrm{diag}(\boldsymbol{u}(k)) & \mathbf{0} \\ \mathbf{0} & \mu\nu\Delta t\boldsymbol{I} \end{pmatrix} \in \mathbb{R}^{8\times 6} \tag{77}$$

$$\bar{\boldsymbol{w}}(k) = \begin{pmatrix} \Delta t\boldsymbol{w}(k) \\ \mu\Delta t\boldsymbol{v}(k) \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, \mathcal{Q}(k)) \in \mathbb{R}^{8} \tag{78}$$

The interpretation of the faults θ_1^a and θ_2^a is associated with a reduction in actuator effectiveness. In these instances, the actual control input generated is less than the nominal value, signifying a diminished capability of the actuators. Conversely, the interpretation of faults θ_1^s, θ_2^s, θ_3^s, and θ_4^s is linked to sensor faults or false injections into the sensor system. In these scenarios, inaccuracies are introduced into the sensor measurements, leading to potential distortions in the perceived state of the system.

4.2. Numerical simulation

In this section, we conduct numerical simulations to investigate the effects of actuator faults, sensor faults, and combined faults on the system. The simulations span a duration of 40 s, with a time step of $\Delta t = 0.01$. This analysis aims to comprehensively assess the impact of different fault scenarios on the behavior and performance of the system over the specified simulation period.

4.2.1. Actuator fault

In this simulation, actuator faults are introduced into the system at $t = 10$ and $t = 20$ s, specifically affecting the control inputs: acceleration $a(t)$ and the rate of the course angle $r(t)$. Instances where actuators experience issues resulting in diminished performance are practical and align with common failure modes. Factors such as broken components, fatigue, or wear are known to contribute to a reduction in the generated input. The initial values for acceleration and course angle rate are set at 1 m/s^2 and 0.1 rad/s, respectively. When the first fault occurs at $t = 10$ s, it leads to a 10% decrease in acceleration. Subsequently, at $t = 20$ s, the second fault results in an additional 10% decrease in acceleration. The impact of these faults on the system's behavior is illustrated in Fig. 6, where deviations from the normal trajectory without faults are observed in both the velocity and course angle profiles. This analysis provides insights into the system's response to actuator faults at different time instances and their cumulative effects on the dynamic behavior.

Fig. 7 illustrates the actual sensor faults alongside their estimations. Notably, the algorithm demonstrates an accurate estimation of the fault magnitudes. The transient period, noticeable at 2 s, indicates the time it takes for the estimation to stabilize. To enhance the convergence rate, optimization can be achieved through tuning parameters, such as the forgetting factor λ and the filtering matrix $\boldsymbol{A}_f$. These parameters play a crucial role in shaping the algorithm's responsiveness and accuracy in fault estimation. Adjusting them appropriately can lead to an improved convergence rate and more precise estimations, further enhancing the algorithm's effectiveness in real-world applications.

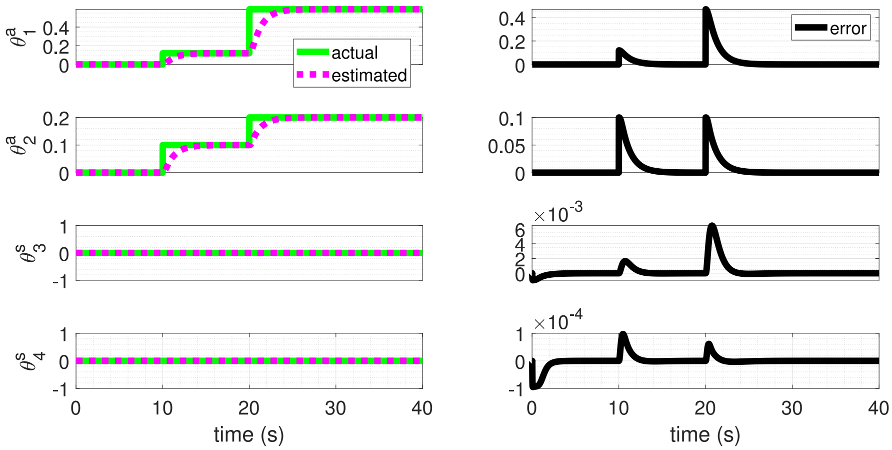

Fig. 7. Estimations of actuator faults (left) and their corresponding errors (right).

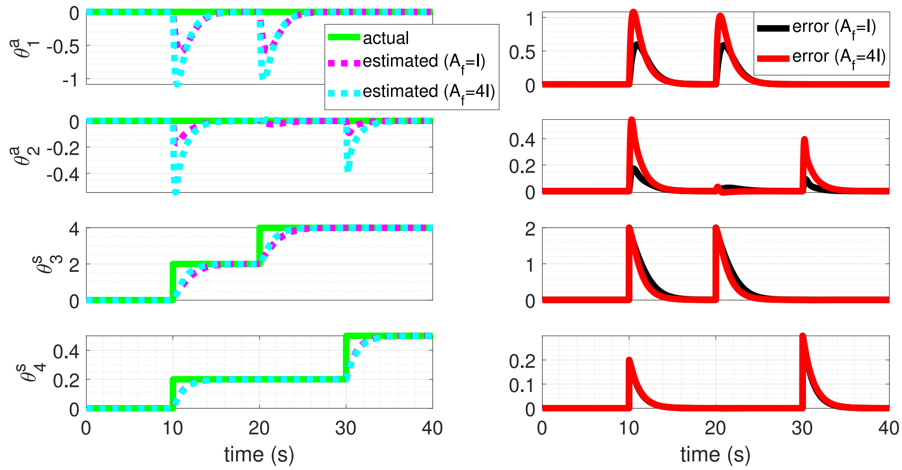

Fig. 8. Estimations of sensor faults (left) and their corresponding errors (right).

4.2.2. Sensor fault

In this simulation, sensor faults affecting the measurements for velocity $U(t)$ and course angle $\chi(t)$ are introduced at $t = 10$, 20, and 30 s. It is important to note that no faults are assumed in the actuator system for this particular scenario. The fault estimation results are presented in Fig. 8, indicating that the algorithm effectively estimates the magnitude of the sensor faults. However, it is evident that the introduction of sensor faults has an impact on the estimation of faults in the actuator system, although the values eventually converge to zero. This influence arises from the fact that our formulation is grounded in the algorithm for actuator fault diagnosis. Addressing this issue could involve refining the filtering matrix $\boldsymbol{A}_f$ to better align with the characteristics of sensor fault estimation, thereby enhancing the algorithm's robustness in scenarios involving mixed faults.

As evident from Fig. 8, the error can be minimized by reducing the eigenvalues of $\boldsymbol{A}_f$. This observation underscores the critical role of the filtering matrix in shaping the performance of the fault estimation algorithm. By decreasing the eigenvalues of $\boldsymbol{A}_f$, the algorithm becomes more responsive to variations in the system, enabling a more accurate estimation of faults. The eigenvalues of $\boldsymbol{A}_f$ essentially influence the rate at which the algorithm adapts to changes in the measured data. A lower set of eigenvalues facilitates a more gradual and controlled adjustment, preventing undue sensitivity to noise or abrupt fluctuations. Thus, tuning the eigenvalues of $\boldsymbol{A}_f$ represents a strategic approach to optimize the fault estimation algorithm, enhancing its robustness and precision in the face of dynamic conditions and mixed fault scenarios. This finding emphasizes the significance of parameter tuning in tailoring fault diagnosis algorithms to specific system characteristics, contributing to their effectiveness in practical applications.

4.2.3. Actuator and sensor fault

In this simulation, we intentionally introduce faults in both the actuator and sensor systems, with these faults occurring at $t = 10$, 20, and 30 s. The outcomes of this mixed-fault scenario are visualized in Fig. 9. Remarkably, the fault estimators converge to the actual values within a short span of 2 s. The convergence rate, a crucial metric in fault estimation, is contingent on the value of the

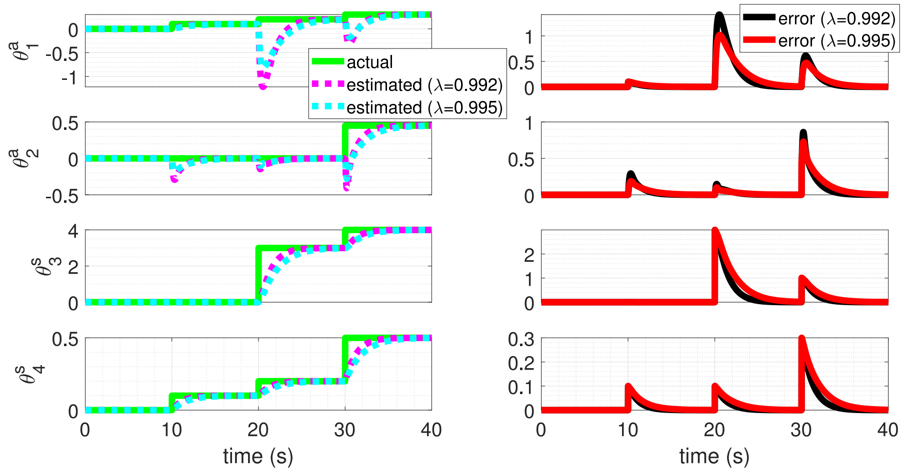

Fig. 9. Estimations of actuator and sensor faults (left) and their corresponding errors (right).

forgetting factor λ. Notably, in this case, increasing λ appears to accelerate the convergence rate. This observation highlights the dynamic relationship between the forgetting factor and the algorithm's ability to adapt swiftly to changes induced by mixed faults. As λ influences the weight assigned to recent measurements, a higher value promotes a more responsive adjustment, enhancing the algorithm's efficiency in accurately estimating faults. This insight underscores the importance of appropriately tuning λ to optimize the performance of fault diagnosis algorithms in scenarios involving concurrent faults in both actuator and sensor systems.

4.3. Drift and noise fault

In the previous section, our focus was solely on the consideration of bias faults, deemed the most prevalent type of fault. However, it is imperative to acknowledge the existence of other fault categories, namely drift and noise, which can also impact the performance of the system. In this section, we consider a numerical simulation that introduces drift and noise faults into the system, providing a more comprehensive assessment of fault scenarios. The simulation unfolds with the injection of faults into the system at $t = 5$ s, simulating real-world conditions where faults may manifest over time. Contrasting with the simplified bias fault scenario, the introduction of drift and noise faults adds layers of complexity to the fault diagnosis process. The subsequent results of state estimation are presented in Fig. 10, offering a visual representation of the filter's performance under these varied fault conditions. It is noteworthy that the filter, even in the presence of drift and noise faults, demonstrates its effectiveness by significantly mitigating the impact of noise on the state estimation. This outcome shows the adaptive nature of the proposed algorithm, showcasing its ability to discern and counteract diverse types of faults. The findings from this simulation contribute to a more understanding of the algorithm's robustness, providing valuable insights into its applicability across a spectrum of fault scenarios encountered in real-world operational contexts.

The outcomes of the fault estimation process are visually depicted in Fig. 11. Notably, the presence of noise becomes evident, exerting an influence on the accuracy of the estimation process. It is apparent that the estimates encounter challenges in accurately capturing the fault under the influence of noise. This observation underscores a crucial aspect of the developed method — its effectiveness in scenarios where faults exhibit constancy or piecewise constancy. The struggle observed in the estimates can be attributed to the inherent characteristics of the algorithm, which is specifically designed to handle constant or piecewise constant faults represented by the parameter vector θ. When faults remain constant, the algorithm excels in accurately estimating them. However, the introduction of noise introduces dynamic fluctuations that pose challenges for the algorithm to discern and precisely capture the fault signature. This insight is valuable for understanding the method's limitations, emphasizing its suitability for scenarios where faults exhibit a more stable pattern. The acknowledgment of these limitations sets the stage for potential refinements or adaptations to enhance the algorithm's performance in scenarios characterized by dynamic and noisy fault conditions. While the algorithm excels in certain fault scenarios, recognizing its constraints fosters a more informed perspective on its applicability, steering future developments and optimizations in fault diagnosis methodologies.

Remark that the accuracy of sensor measurements significantly impacts the performance of fault diagnosis. Precise sensor data is crucial for accurately detecting and diagnosing faults within a system. Inaccurate measurements due to bias, drift, and noise, can lead to erroneous estimations, potentially masking the true state of the system and causing the fault diagnosis algorithm to misidentify or overlook faults. This can result in delays in fault detection, incorrect fault isolation, and ineffective fault correction strategies, ultimately compromising the reliability and safety of the system. Therefore, high-quality, accurate sensor measurements are essential to ensure the robustness and effectiveness of fault diagnosis processes, especially in real-time applications where timely and precise fault detection is critical.

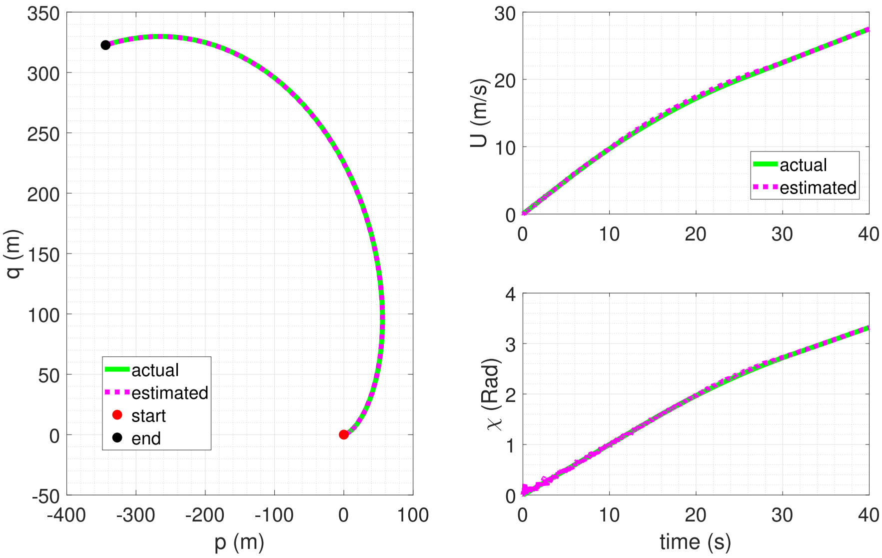

Fig. 10. Estimations of actuator and sensor faults (left) and their corresponding errors (right).

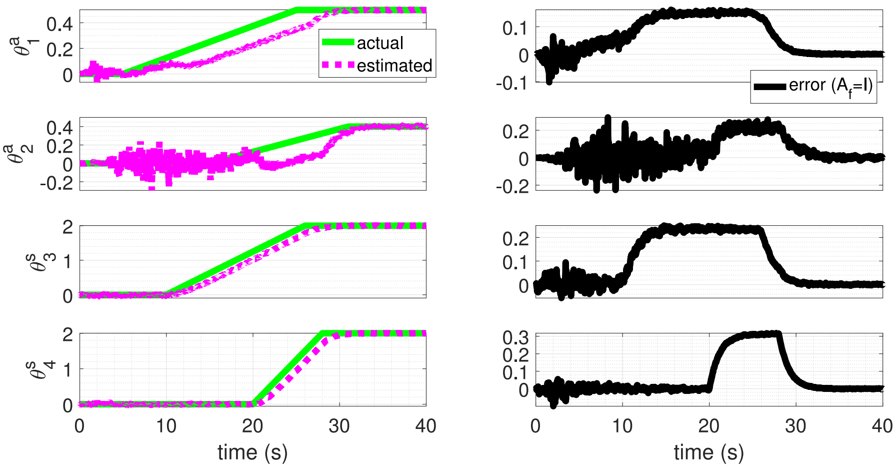

Fig. 11. Estimations of actuator and sensor faults (left) and their corresponding errors (right).

4.4. Comparison with existing methods

The objective of this section is to conduct a comparative analysis between the proposed adaptive extended Kalman filter (AEKF) and other existing methods, such as the conventional extended Kalman filter (CEKF) and the unscented Kalman filter (UKF). This comparative analysis will shed light on the effectiveness and applicability of our proposed methodology in addressing joint sensor and actuator fault scenarios. The CEKF and UKF typically follow a standard procedure when addressing joint sensor and actuator fault problems. In this case, the idea is to transform the original joint state and parameter estimation problem:

$$\xi(k) = \mathcal{A}(k)\xi(k-1) + \boldsymbol{f}(\xi(k-1)) + \mathcal{B}(k)\boldsymbol{u}(k) + \bar{\boldsymbol{\Psi}}(k)\boldsymbol{\theta} + \bar{\boldsymbol{w}}(k) \tag{79}$$

$$\mathcal{Y}(k) = \mathcal{C}(k)\xi(k) \tag{80}$$

into the following augmented formulation:

$$\begin{pmatrix}\xi(k)\\ \boldsymbol{\theta}(k)\end{pmatrix} = \begin{pmatrix}\mathcal{A}(k) & \bar{\boldsymbol{\Psi}}(k)\\ \boldsymbol{0} & \boldsymbol{I}\end{pmatrix}\begin{pmatrix}\xi(k-1)\\ \boldsymbol{\theta}(k-1)\end{pmatrix} + \begin{pmatrix}\boldsymbol{f}(\xi(k-1))\\ \boldsymbol{0}\end{pmatrix} + \begin{pmatrix}\mathcal{B}(k)\\ \boldsymbol{0}\end{pmatrix}\boldsymbol{u}(k) + \begin{pmatrix}\bar{\boldsymbol{w}}(k)\\ \boldsymbol{0}\end{pmatrix} \tag{81}$$

$$\mathcal{Y}(k) = \begin{pmatrix}\mathcal{C}(k) & \boldsymbol{0}\end{pmatrix}\begin{pmatrix}\xi(k)\\ \boldsymbol{\theta}(k)\end{pmatrix} \tag{82}$$

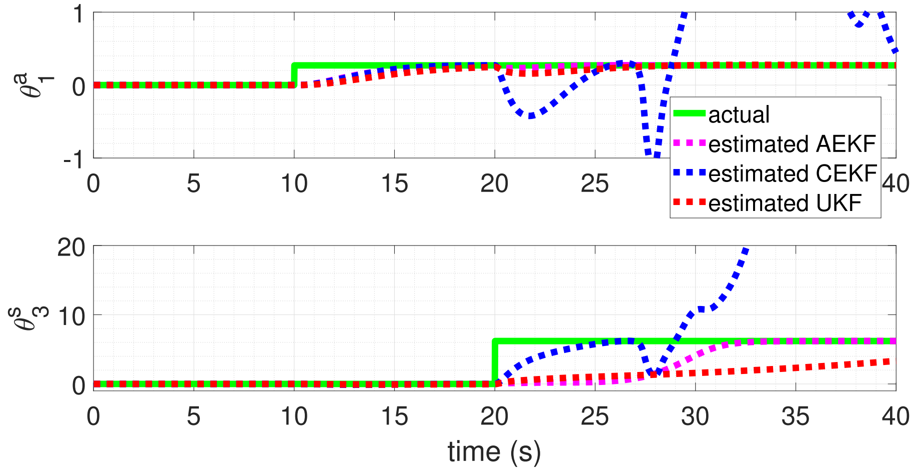

Fig. 12. Comparison between the proposed method (AEKF), conventional EKF, and UKF.

To guarantee the stability of the Kalman filter, it is essential for the augmented system to exhibit both uniform complete observability and uniform complete controllability with respect to the state noise. In our specific scenario, it becomes evident that the augmented system lacks uniform complete controllability concerning the state noise term $\bar{\boldsymbol{w}}(k)$. This deficiency arises from the fact that the augmented states $\begin{pmatrix}\xi(k) & \theta(k)\end{pmatrix}^\intercal$ remain entirely uncontrollable with respect to the influence exerted by the state noise $\bar{\boldsymbol{w}}(k)$. In other words, the augmented states are not subject to any control or correction by the state noise term, thereby highlighting a limitation in the controllability aspect of the augmented system.

In the CEKF algorithm, the estimation of both state and parameters, denoted as $\begin{pmatrix}\hat{\xi}(k) & \hat{\theta}(k)\end{pmatrix}^\intercal$, is achieved through the implementation of the standard extended Kalman filter algorithm, as expressed in Eqs. (44)–(47). Notably, in this process, the conventional state transition matrix $\mathcal{A}(k)$ is substituted with the Jacobian matrix of the nonlinear function, specifically $\begin{pmatrix}\boldsymbol{f}(\xi(k-1)) & \mathbf{0}\end{pmatrix}^\intercal$. This modification accounts for the dynamic evolution of the system and enables an effective update of the state and parameter estimates within the CEKF framework. The inclusion of the Jacobian matrix enhances the algorithm's ability to handle nonlinearity in the system dynamics, providing a more accurate and refined estimation.

In the UKF algorithm, the selection of sigma points, denoted as χ_i for $i \in 1, \dots, 2n+1$, pertaining to the state and parameter estimates $\begin{pmatrix}\hat{\xi}(k) & \hat{\theta}(k)\end{pmatrix}^\intercal$, follows certain rules:

$$\chi_i(k) = \begin{cases} \begin{pmatrix}\hat{\xi}(k) & \hat{\theta}(k)\end{pmatrix}^\intercal, & \text{if} \quad i = 1, \\ \begin{pmatrix}\hat{\xi}(k) & \hat{\theta}(k)\end{pmatrix}^\intercal + (\sqrt{(n-\alpha)\mathbf{P}_i(k)}), & \text{if} \quad i = 2, \dots, n+1, \\ \begin{pmatrix}\hat{\xi}(k) & \hat{\theta}(k)\end{pmatrix}^\intercal - (\sqrt{(n-\alpha)\mathbf{P}_i(k)}), & \text{if} \quad i = n+2, \dots, 2n+1, \end{cases} \tag{83}$$

where n represents the number of states and α serves as a scaling parameter determining the spread of the sigma points. This parameter is defined as $\alpha = (n+\beta)\gamma^2 - n$, where γ is a small positive value dictating the distribution parameter of the sigma points around $\begin{pmatrix}\hat{\xi}(k) & \hat{\theta}(k)\end{pmatrix}^\intercal$. Additionally, β functions as an extra scaling parameter, commonly set to zero.

The tuning parameters for the AEKF, CEKF, and UKF have been fine-tuned using Particle Swarm Optimization (PSO), a metaheuristic optimization technique. The comparative performance of these filters is illustrated in Fig. 12. The analysis of the results reveals notable distinctions among the filters. Both the AEKF and UKF exhibit convergence towards the actual faults, implying their effectiveness in fault estimation. However, a noteworthy observation is that the CEKF diverges, indicating a departure from the actual fault states. This outcome aligns with the inherent challenge associated with augmented states, which can render them entirely uncontrollable in certain scenarios. In contrast, the UKF, while ultimately converging to the correct fault states, exhibits a relatively slow convergence rate. This suggests that, despite its ability to accurately estimate faults, the UKF may require more computational time or measurements to achieve convergence compared to the AEKF. These insights into the performance characteristics of the filters shed light on their respective strengths and limitations in fault estimation applications.

4.5. Experimental validation

In this section, we implement the estimation algorithm utilizing data acquired from the Otter ASV. The objective is to assess the algorithm's performance under realistic conditions by injecting faults into the Otter's control systems. Specifically, for actuator faults, we simulate failure scenarios by reducing the control gains, thereby decreasing the nominal control input into the Otter. This approach emulates the impact of actual actuator failures on the ASV's control mechanisms. Additionally, for the sensor system, faults are introduced in the form of biases. These biases introduce inaccuracies in the sensor measurements, mimicking real-world sensor

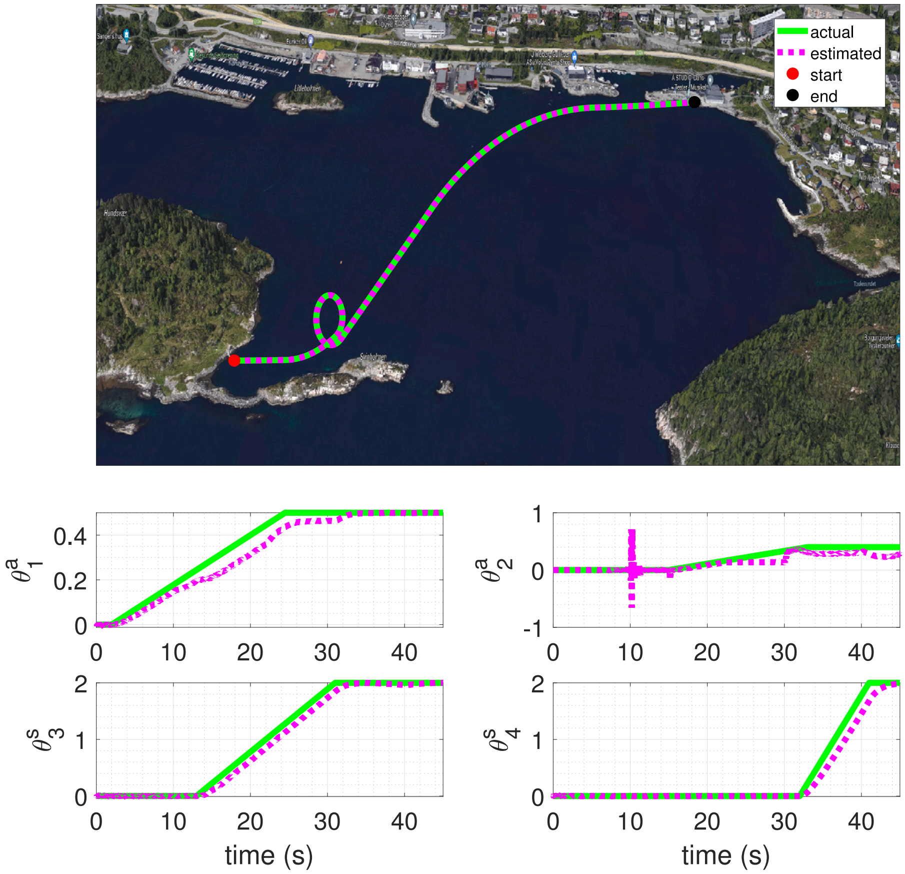

Fig. 13. Experimental result of fault diagnosis using data from the Otter ASV.

faults. By implementing faults in both the actuator and sensor systems, we aim to evaluate the algorithm's robustness and accuracy in diagnosing concurrent faults. This real-world simulation not only enhances the algorithm's applicability but also provides valuable insights into its effectiveness under dynamic conditions. The approach of simulating faults in a controlled environment using data from the Otter ASV ensures a practical and representative evaluation of the estimation algorithm's performance in the context of autonomous maritime operations.

In this scenario, the Otter embarks on a journey from the island Svinholmen, traversing the Borgundfjorden. The travel spans approximately 45 s, during which we record the vessel's position, velocity, and course angle. At $t = 9$ s, actuator faults are intentionally introduced, simulating a scenario where the Otter's propulsion system experiences anomalies. Subsequently, at $t = 20$ s, sensor faults are introduced, creating inaccuracies in the measurements of position, velocity, and course angle. The outcomes of this dynamic scenario are visually depicted in Fig. 13. Notably, the fault diagnosis algorithm adeptly estimates all introduced faults with precision. This observation underscores the algorithm's effectiveness in accurately identifying and diagnosing both actuator and sensor faults, even under the dynamic conditions of a maritime journey. The real-time assessment of faults during the Otter's transit provides valuable insights into the algorithm's performance in a practical maritime setting, demonstrating its robustness and applicability for fault diagnosis in autonomous surface vehicles.

Remark that in the area of remote operations, the presence of signal delays and the introduction of additional noise during transmission pose potential challenges. These factors, stemming from the inherent nature of remote communication, may influence the accuracy of sensor measurements crucial for our proposed methodology, which leverages the Kalman filter. However, our experimental observations reveal that the impact of signal delays and added noise on the performance of our adaptive extended Kalman filter is not significantly pronounced. The adaptive nature of our filter allows it to effectively mitigate the effects of these challenges, showcasing its robustness in real-world remote operation scenarios.

5. Conclusion and future works

In this paper, we introduced a novel unified hybrid approach algorithm designed for sensor and actuator fault diagnosis. This algorithm utilizes a versatile model capable of detecting faults in both actuator and sensor systems. The key innovations in this work involve the transformation of sensor faults into the state space model and the implementation of an adaptive extended

Kalman filter algorithm, which deviates from conventional Kalman filter approaches. Our algorithms incorporate tuning parameters, providing flexibility for adjustments that enhance the convergence rate. Supported by numerical simulations and experimental validations using real data, our approach showcases proficiency in accurately estimating fault magnitudes. The results demonstrate the algorithm's robustness across simulated scenarios and practical applications with real-world data, affirming its potential for deployment in autonomous systems. It is noteworthy that higher fidelity models can be employed for more precise fault localization, tailoring the algorithm to specific system intricacies. Future works will focus on refining and expanding upon the presented algorithms. This includes investigating the adaptability of the algorithms to diverse system architectures and fault types, optimizing tuning parameters for specific applications, and using high-fidelity model of the systems. Moreover, research efforts could explore the integration of machine learning techniques to enhance fault detection capabilities, further improving the algorithm's performance in complex operational environments. The presented unified hybrid approach lays the foundation for advancements in fault diagnosis methodologies, offering promising avenues for continued research and development in this field.

CRediT authorship contribution statement

Agus Hasan: Writing – original draft, Methodology, Data curation, Conceptualization. **Pierluigi Salvo Rossi:** Writing – original draft, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve readability and language. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Acknowledgment

This work is funded by Equinor ASA with project number 101593114.

References

[1] K. Brunnström, E. Dima, T. Qureshi, M. Johanson, M. Andersson, M. Sjöström, Latency impact on quality of experience in a virtual reality simulator for remote control of machines, Signal Process., Image Commun. 89 (2020) 116005.
[2] E.E. Ambarita, A. Karlsen, O. Osen, A. Hasan, Towards fully autonomous floating offshore wind farm operation & maintenance, Energy Rep. 9 (2023) 103–108.
[3] C. Liu, P. Jiang, W. Jiang, Web-based digital twin modeling and remote control of cyber-physical production systems, Robot. Comput.-Integr. Manuf. 64 (2020) 101956.
[4] A. Haghshenas, A. Hasan, O. Osen, E. Mikalsen, Predictive digital twin for offshore wind farms, Energy Inform. 6 (2023) 1–26.
[5] R. van Dinter, B. Tekinerdogan, C. Catal, Predictive maintenance using digital twins: A systematic literature review, Inf. Softw. Technol. 151 (2022) 107008.
[6] M.T. Qaiser, J. Ejaz, O. Osen, A. Hasan, Digital twin-driven energy modeling of hywind tampen floating wind farm, Energy Rep. 9 (2023) 284–289.
[7] E. Kandemir, J. Liu, A. Hasan, Digital twin-driven dynamic repositioning of floating offshore wind farms, Energy Rep. 9 (2023) 208–214.
[8] R. Bhagavathi, D.K.M. Kufoalor, A. Hasan, Digital twin-driven fault diagnosis for autonomous surface vehicles, IEEE Access 11 (2023) 41096–41104.
[9] J. Pan, L. Qu, K. Peng, Sensor and actuator fault diagnosis for robot joint based on deep CNN, Entropy 23 (6) (2021).
[10] M. Kordestani, M. Saif, M.E. Orchard, R. Razavi-Far, K. Khorasani, Failure prognosis and applications—A survey of recent literature, IEEE Trans. Reliab. 70 (2) (2021) 728–748.
[11] H. Darvishi, D. Ciuonzo, E.R. Eide, P. Salvo Rossi, Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture, IEEE Sens. J. 21 (4) (2021) 4827–4838.
[12] H. Darvishi, D. Ciuonzo, P. Salvo Rossi, A machine-learning architecture for sensor fault detection, isolation, and accommodation in digital twins, IEEE Sens. J. 23 (3) (2023) 2522–2538.
[13] D. Gonzalez-Jimenez, J. del Olmo, J. Poza, F. Garramiola, P. Madina, Data-driven fault diagnosis for electric drives: A review, Sensors 21 (12) (2021).
[14] L. Gómez-Coronel, I. Santos-Ruiz, L. Torres, F.-R. López-Estrada, S. Gómez-Peñate, E. Escobar-Gómez, Digital twin of a hydraulic system with leak diagnosis applications, Processes 11 (10) (2023).
[15] Y. Lei, B. Yang, X. Jiang, F. Jia, N. Li, A.K. Nandi, Applications of machine learning to machine fault diagnosis: A review and roadmap, Mech. Syst. Signal Process. 138 (2020) 106587.
[16] H. Darvishi, D. Ciuonzo, P. Salvo Rossi, Deep recurrent graph convolutional architecture for sensor fault detection, isolation, and accommodation in digital twins, IEEE Sens. J. 23 (23) (2023) 29877–29891.
[17] C. Molnar, G. Casalicchio, B. Bischl, Interpretable machine learning – A brief history, state-of-the-art and challenges, in: I. Koprinska, M. Kamp, A. Appice, C. Loglisci, L. Antonie, A. Zimmermann, R. Guidotti, O. Özgöbek, R.P. Ribeiro, R. Gavaldà, J.a. Gama, L. Adilova, Y. Krishnamurthy, P.M. Ferreira, D. Malerba, I. Medeiros, M. Ceci, G. Manco, E. Masciari, Z.W. Ras, P. Christen, E. Ntoutsi, E. Schubert, A. Zimek, A. Monreale, P. Biecek, S. Rinzivillo, B. Kille, A. Lommatzsch, J.A. Gulla (Eds.), ECML PKDD 2020 Workshops, 2020, pp. 417–431.

[18] J. Dalzochio, R. Kunst, E. Pignaton, A. Binotto, S. Sanyal, J. Favilla, J. Barbosa, Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges, Comput. Ind. 123 (2020) 103298.

[19] A. Hasan, M. Tahavori, H.S. Midtiby, Model-based fault diagnosis algorithms for robotic systems, IEEE Access 11 (2023) 2250–2258.

[20] M. Schmid, E. Gebauer, C. Hanzl, C. Endisch, Active model-based fault diagnosis in reconfigurable battery systems, IEEE Trans. Power Electron. 36 (3) (2021) 2584–2597.

[21] A. Hasan, T.A. Johansen, Model-based actuator fault diagnosis in multirotor UAVs, in: 2018 International Conference on Unmanned Aircraft Systems, ICUAS, 2018, pp. 1017–1024.

[22] T. Ekanayake, D. Dewasurendra, S. Abeyratne, L. Ma, P. Yarlagadda, Model-based fault diagnosis and prognosis of dynamic systems: a review, Procedia Manuf. 30 (2019) 435–442.

[23] H. Habibi, I. Howard, S. Simani, Reliability improvement of wind turbine power generation using model-based fault detection and fault tolerant control: A review, Renew. Energy 135 (2019) 877–896.

[24] Y. Wilhelm, P. Reimann, W. Gauchel, B. Mitschang, Overview on hybrid approaches to fault detection and diagnosis: Combining data-driven, physics-based and knowledge-based models, Procedia CIRP 99 (2021) 278–283.

[25] A. Slimani, P. Ribot, E. Chanthery, N. Rachedi, Fusion of model-based and data-based fault diagnosis approaches, IFAC-PapersOnLine 51 (24) (2018) 1205–1211.

[26] H. Jin, Z. Gao, Z. Zuo, Z. Zhang, Y. Wang, A. Zhang, A combined model-based and data-driven fault diagnosis scheme for lithium-ion batteries, IEEE Trans. Ind. Electron. (2023) 1–11.

[27] R.E. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME J. Basic Eng. 82 (Series D) (1960) 35–45.

[28] Q. Zhang, Adaptive Kalman filter for actuator fault diagnosis, Automatica 93 (2018) 333–342.