DOCTORAL THESIS

Data-Driven Fault-Resilient Cross-Layer Sensor Network Architecture

Lauri Vihman

TALLINNA TEHNIKAÜLIKOOL
TALLINN UNIVERSITY OF TECHNOLOGY
TALLINN 2024

TALLINN UNIVERSITY OF TECHNOLOGY
DOCTORAL THESIS
7/2024

Data-Driven Fault-Resilient Cross-Layer Sensor Network Architecture

LAURI VIHMAN

TALTECH
PRESS

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies
Department of Computer Systems

The dissertation was accepted for the defence of the degree of Doctor of Philosophy on 8th February 2024

Supervisor: Professor Dr. Jaan Raik,
Department of Computer Systems, School of Information Technologies,
Tallinn University of Technology,
Tallinn, Estonia

Co-supervisor: Professor Dr. Maarja Kruusmaa,
Department of Computer Systems, School of Information Technologies,
Tallinn University of Technology,
Tallinn, Estonia

Opponents: Professor Dr. Fausto Ferreira,
Faculty of Electrical Engineering and Computing,
University of Zagreb,
Zagreb, Croatia

Professor Dr. Marco Ottavi,
University of Rome Tor Vergata / University of Twente,
Rome, Italy / Enschede, Netherlands

Defence of the thesis: 26th February 2024, Tallinn

Declaration:
Hereby I declare that this doctoral thesis, my original investigation and achievement, submitted for the doctoral degree at Tallinn University of Technology, has not been submitted for any academic degree elsewhere.

Lauri Vihman

signature

Copyright: Lauri Vihman, 2024
ISSN 2585-6898 (publication)
ISBN 978-9916-80-109-3 (publication)
ISSN 2585-6901 (PDF)
ISBN 978-9916-80-110-9 (PDF)
DOI https://doi.org/10.23658/taltech.7/2024
Printed by Koopia Niini & Rauam

Vihman, L. (2024). *Data-Driven Fault-Resilient Cross-Layer Sensor Network Architecture* [TalTech Press]. https://doi.org/10.23658/taltech.7/2024

TALLINNA TEHNIKAÜLIKOOL
DOKTORITÖÖ
7/2024

Andmepõhine tõrkekindel kihtideülene sensorvõrgu arhitektuur

LAURI VIHMAN

KIRJASTUS

Contents

List of Publications 7

Author's Contributions to the Publications 8

Abbreviations 9

1 Introduction 10
  1.1 Motivation 10
  1.2 Problem Formulation 11
  1.3 Contributions 13
  1.4 Thesis Structure 14

2 Related Works 15
  2.1 Motivation 15
  2.2 Method 15
  2.3 Results 15
    2.3.1 Fault Tolerance Tasks 15
    2.3.2 Comparative Analysis 17
    2.3.3 Open Research Issues 25
  2.4 Chapter Summary 25

3 Preliminaries: Sensors, Sensor Network Framework, Data Layers, and Case Study Scenarios 26
  3.1 Motivation 26
  3.2 Sensors Used for Experiments 26
    3.2.1 Underwater Sensor Network Installation 26
    3.2.2 Indoor Sensor Network Installation 27
  3.3 Implemented Software framework for sensor networks 28
    3.3.1 Overview of Related Works 28
    3.3.2 High-Level Software Architecture of the Framework 29
  3.4 Proposed Layered Data Model 30
  3.5 Chapter Summary 32

4 Data-Driven Method for Processing Raw Sensor Data 33
  4.1 Motivation 33
  4.2 Applicability in the context of USN's 33
  4.3 Data Collection: Smart Building Case Study 34
  4.4 CO_2 Concentration Correction Results 35
  4.5 Chapter Summary 37

5 Cross-Layer Fault Management in Data Processing 38
  5.1 Motivation 38
  5.2 Fault Detection and Diagnosis at Different Data Layers 38
  5.3 Results 40
    5.3.1 Graceful Data Degradation Enabled by Fault Management 40
    5.3.2 Incident 1: Fault detection at the Raw Layer 41
    5.3.3 Incident 2: Fault Detection at the Processed Layer 43
  5.4 Chapter Summary 45

6 Fault Management in Data Aggregation 46
  6.1 Motivation 46
  6.2 Methods 46
    6.2.1 Uncertainty from the Difference of Measurement and Estimation .. 46
  6.3 Results 48
    6.3.1 Flow Obstruction Experiment 48
    6.3.2 Harbor Experiment 49
  6.4 Chapter Summary 51

7 Conclusions 52
  7.1 Summary 52
  7.2 Limitations and Future work 53

List of Figures 54

List of Tables 55

References 56

Acknowledgements 69

Abstract 70

Kokkuvõte 71

Appendix 1 73

Appendix 2 83

Appendix 3 107

Appendix 4 117

Appendix 5 125

Curriculum Vitae 139

Elulookirjeldus 141

List of Publications

The present Ph.D. thesis is based on the following publications that are referred to in the text by Roman numbers.

- I L. Vihman, M. Kruusmaa, and J. Raik. Data-Driven Cross-Layer Fault Management Architecture for Sensor Networks. In *2020 16th Eur. Dependable Comput. Conf.*, pages 33–40. IEEE, sep 2020
- II L. Vihman, M. Kruusmaa, and J. Raik. Systematic Review of Fault Tolerant Techniques in Underwater Sensor Networks. *Sensors*, 21(9):3264, may 2021
- III L. Vihman, T. M. Parts, H. K. Aljas, M. Thalfeldt, and J. Raik. Algorithms for online CO2 baseline correction in intermittently occupied rooms. In *18th Heal. Build. Eur. Conf.*, Aachen, Germany, 2023. Emerald
- IV L. Vihman and J. Raik. Adaptive Kalman Filter Based Data Aggregation in Fault-Resilient Underwater Sensor Networks. In *2023 24th Int. Conf. Digit. Signal Process.*, pages 1–5. IEEE, jun 2023
- V M. Egerer, A. Ristolainen, L. Piho, L. Vihman, and M. Kruusmaa. Hall effect sensor-based low-cost flow monitoring device: Design and validation. *IEEE Sensors Journal*, pages 1–1, 2024

Author's Contributions to the Publications

I In I, I was the main author, wrote the software framework, conducted the experiments, proposed the concept, wrote the simulations, prepared the figures, wrote the manuscript.

II In II, I was the main author, conceptualized the topic, implemented methodology, software, validation, formal analysis and investigation, prepared original draft and edited it, created visualization.

conducted the systematic searches and gathered the results, read, categorized, and analyzed the papers, prepared the figures, and wrote the manuscript.

III In III, I was the main author, developed and tested the algorithm, prepared the figures, prepared and edited the manuscript draft.

IV In IV, I was the main author, wrote the software framework for the experiments, conducted the experiments, developed the algorithms, analyzed the results, prepared the figures, wrote the manuscript.

V In V, I was the fourth author, designed the software framework used, wrote the software components, integrated the cloud services, and helped conduct the experiments.

Abbreviations

ADV	Acoustic Doppler Velocimeter
AWS	Amazon Web Services
CA	Certificate Authority
CDK	Cloud Development Kit
DCV	Demand Controlled Ventilation
FFT	Fast Fourier Transform
FIFO	First In, First Out
FT	Fault Tolerance
IaC	Infrastructure as Code
IAQ	Indoor Air Quality
IMU	Inertial Measurement Unit
IoT	Internet of Things
IoUT	Internet of Underwater Things
IP	Internet Protocol
KF	Kalman Filter
LoRa	Long Range (a radio communication protocol)
MQTT	MQ Telemetry Transport (a lightweight network protocol)
ppm	parts per million
PSD	Power Spectral Density
RH	Relative Humidity
SBC	Single Board Computer
SSL	Secure Sockets Layer
USN	Underwater Sensor Network
VPN	Virtual Private Network

1 Introduction

1.1 Motivation

Underwater Sensor Networks (USNs) have become widespread and are being deployed in a wide range of applications, ranging from harbor security to monitoring underwater pipelines and fish farms. Due to the fact that USNs often operate in an extremely harsh environment, and many of their applications are safety critical, it is imperative to develop techniques enabling these networks to tolerate faults. Moreover, USNs face many challenges that are not present in terrestrial networks, such as the virtual inapplicability of the wireless radio communication under water and the limitations of the acoustic means. The challenges of the underwater environment arise from environmental conditions - high pressure, low temperature, high turbulence, exposure to aquatic life, and engineering constraints - limited communication, costly maintenance, and energy constraints.

It is important to stress that the underwater environment is mostly different from terrestrial conditions, in the sense of additional and more fatal hazards, such as increased pressure and the danger of flooding, as well as added difficulty of communication and physical access. Some communication media, such as radio signals, are not applicable underwater. Furthermore, falling temperatures with increasing depth may affect the equipment's operation and reliability. The challenges of fault management in sensor networks include:

- Reliability and accuracy in sensor networks are important because their data is often used for critical decision making. Faults in sensor networks may lead to inaccurate and unreliable data.

- System Resilience: Sensor networks often operate in challenging environments, where failures can occur due to various factors such as environmental conditions, hardware malfunctions, or communication issues. Fault management techniques improve the resilience of the sensor network by enabling fault detection, localization, and recovery mechanisms. By quickly identifying faults and implementing appropriate actions, the system can continue to function or gracefully degrade, minimizing disruptions and ensuring continuous operation.

- Resource Optimization: Sensor networks often operate under resource-constrained conditions, including limited energy, bandwidth, and processing capabilities. Effective fault management helps optimize the utilization of these resources. By promptly detecting faults, the network can dynamically reconfigure, redistribute tasks, or activate backup resources, thus minimizing unnecessary resource consumption and prolonging the network's overall lifespan.

- Cost Reduction: Fault management plays an important role in reducing the maintenance and repair costs associated with sensor networks. By proactively detecting faults and addressing them before they escalate, the need for extensive repairs or replacements can be minimized. Furthermore, fault management techniques enable targeted maintenance activities, focusing only on affected components, rather than performing routine maintenance on the entire network, thus optimizing resource allocation and reducing costs.

- Safety and Critical Applications: Sensor networks are often deployed in safety-critical applications such as environmental monitoring, industrial control systems, or healthcare. Fault management becomes particularly important in these scenarios, as faulty data or system failures can have severe consequences, including safety hazards or compromised operational efficiency. By implementing effective fault management strategies, the integrity and reliability of the sensor network can be ensured, mitigating potential risks and ensuring the safety of the overall system.

Overall, fault management is crucial in sensor networks to maintain reliability, accuracy, system resilience, resource optimization, cost reduction, and safety. By proactively managing faults, sensor networks can operate efficiently, provide accurate data, and fulfill their intended purposes in a wide range of applications.

Hardware faults may occur for various reasons, including:

- Manufacturing Defects: During the manufacturing process, hardware components can have defects or errors introduced. These defects may arise from issues in the fabrication process, assembly errors, or faulty components. Manufacturing defects can lead to hardware malfunctions or failures that cause faults.
- Wear and Tear: Continuous usage, environmental conditions, and aging can cause wear and tear in hardware components. Over time, this can lead to material degradation, deterioration of electrical connections, or mechanical failures. Such wear and tear can manifest as hardware faults.
- Environmental Factors: Hardware can be exposed to various environmental factors that can contribute to faults. Extreme temperatures, humidity, vibrations, electromagnetic interference, or exposure to corrosive substances can affect the performance and reliability of hardware components. These environmental factors may cause electrical shorts, component degradation, or physical damage, leading to faults.
- Power Supply Issues: Fluctuations or disruptions in the power supply can affect the stability and functionality of hardware components. Power surges, voltage drops, or sudden power outages can cause transient or permanent faults in the hardware. Insufficient power supply or improper grounding can also contribute to faults.
- Human Errors: Human errors during installation, maintenance, or handling of hardware can introduce faults. Improper installation, incorrect configurations, or accidental physical damage can impact the performance and reliability of hardware components. Incorrect handling, such as mishandling sensitive connectors or inserting components in the wrong orientation, can also result in faults.

Due to manufacturing advances the number of hardware faults is predicted not to disappear but to increase in the future [127]. One way to cope with faults is to accept imperfect devices to fail and compensate failures at higher levels in the system stack [34], tolerating faults across layers involving circuit design, firmware, operating system, applications, etc. Cross-layer fault tolerant systems have potential to implement reliable, high-performance and energy-efficient solutions without overwhelming costs [21] by distributing the responsibilities of tolerating faults across multiple layers [154].

1.2 Problem Formulation

The research questions addressed in this thesis are the following ones:

RQ1. What are the research caps in cross-layer fault tolerant underwater sensor networks?

RQ2. How can data management help estimate quality classes?

RQ3. How can the data and essential intrinsic functionality of sensor networks be used for cross-layer fault management?

RQ4. How to validate the quality and reliability of aggregated data, considering the integration of data from multiple sensors, and incorporating techniques to handle faults or anomalies during aggregation?

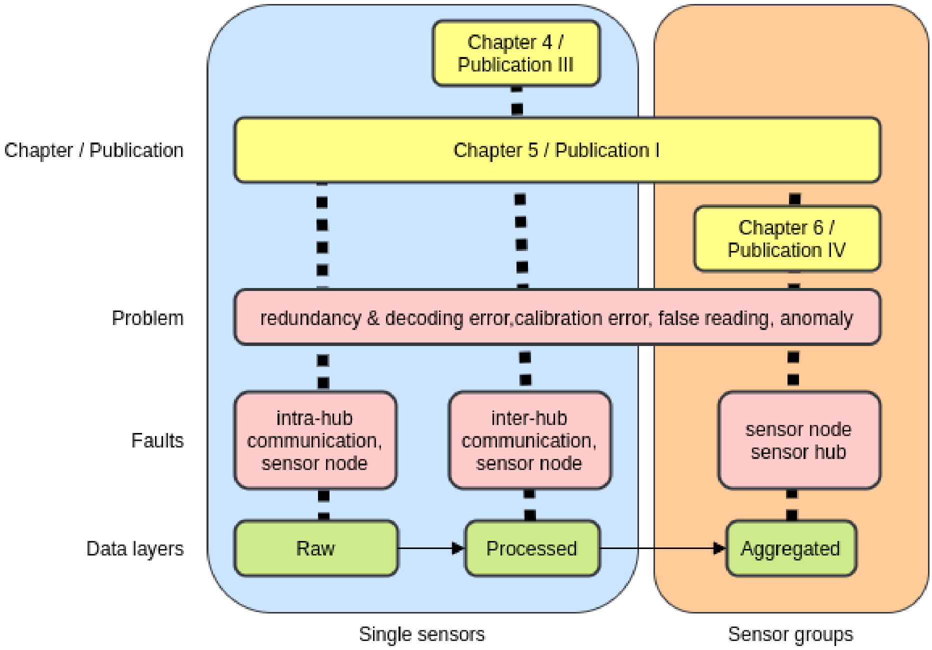

Chapter 4 / Publication III
Chapter / Publication
Chapter 5 / Publication I
Chapter 6 / Publication IV
Problem
redundancy & decoding error,calibration error, false reading, anomaly
Faults
intra-hub communication, sensor node
inter-hub communication, sensor node
sensor node sensor hub
Data layers
Raw
Processed
Aggregated
Single sensors
Sensor groups

Figure 1: Fault management in data layers presented in current thesis.

While usually cross-layer fault management is viewed from the perspective of system layers [34], the systems can be developed in multiple ways, functionalities implemented in different system layers, and fault management does not have to be coupled to a specific implementation.

Thus, current work looks at cross-layer fault management from the perspective of sensor data. Data are an essential functionality of the sensor network and have built-in redundancy properties. The author proposes data layers, Raw, Processed, and Aggregated, to fulfill the cross-layer fault tasks. Figure 1 shows the defined data layers and their relations to fault sources, problems, corresponding published publications by the author of the current thesis and chapters covering specific topics. The data-driven approach utilizes the inherent redundancy within the sensor network as opposed to applying dedicated error monitors and/or duplicated hardware resources.

The problems that the author addresses in this thesis are redundancy, decoding, and calibration errors, false readings, and data anomalies. The faults, for instance in intra- and inter-hub communications, and sensor nodes, are typically detected when advancing to the upper data layers.

The thesis work is divided as shown in the figure between single sensors and sensor groups. Publication III (discussed in Chapter 4) covers faults by processing single sensor data in Raw data layers. Next, Publication I (Chapter 5) discusses the single sensors faults on procesed data layer and proposes Data-driven cross Layer Fault Management concept. Finally, Publication IV (Chapter 6) is covering the faults in sensor groups and Aggregated data layer. Publication V is discussed in Chapter 3 and is not presented on the Figure 1.

1.3 Contributions

The main contribution of this thesis is **proposing a data-driven and cross-layer resilient architecture for sensor networks, where instead of system stack layers, data layers are applied to the detection and diagnosis of faults**. Specific contributions categorized by chapters are as follows.

- Conducting the first survey of fault-tolerant, particularly cross-layer fault-tolerant, techniques in USNs (in Chapter 2, addresses research question RQ1)
    - Introducing a new taxonomy of the Fault Tolerance tasks for categorizing fault-tolerant techniques for USNs;
    - Presenting a comprehensive, categorized list of 127 articles of works applicable in fault-tolerant USN design and deployment;
    - Listing the open research issues within the focused area;
- Proposing a data-driven method for processing the sensor data to improve the measurement accuracy and improve the estimation of quality classes based on that. (in Chapter 4, RQ2);
    - Developing and applying an baseline correction algorithm for in-door CO_2 base level based on CO2 data logged during 6 months from 56 rooms.
    - Improving quality estimation by almost 3 times by applying the baseline correction algorithm for more than 1000ppm CO_2 threshold in rooms.
- Proposing three data layers for cross-layer fault management - raw, processed, aggregated; (in Chapter 5, RQ3)
    - The data-driven cross-layer fault management allows improving the sensor group measurement accuracy by 35% in case of a single sensor fault and nearly twofold in case of double sensor fault.
- Proposing a data-driven method for aggregating sensor data to improve quality (in Chapter 6, RQ4).
    - Incorporating different sources of sensor uncertainty by including the time series measurements' difference and age/latency uncertainty for adapting a Kalman Filter to compensate incorrect readings for more efficient state prediction;
    - Proposing nonlinear, parabolic and sigmoid, sensor uncertainty functions from the residual difference for the latency and difference based Adaptive Kalman Filter techniques, respectively.
- Applying and evaluating the proposed techniques in natural environments with extremely unreliable sensor readings(in Chapters 3.3, 5, 6)

1.4 Thesis Structure

The rest of the thesis is organized as follows. First, Chapter 2 gives a review of related works and the current state of the art on the current topic, as well as introduce the taxonomy of fault management tasks for sensor networks. Chapter 3 introduces the software framework architecture developed by the author for the current thesis. Next, Chapter 4 covers the initial fault management processing in a sensor network based on building in-door climate measurement. Following, Chapter 5 covers the processing of the faults in an underwater sensor network detected on Raw and Processed data layers. Chapter 6 explains fault filtering in the Aggregated data layer for sensor groups. Finally, Chapter 7 concludes the thesis.

2 Related Works

2.1 Motivation

The main goal of this chapter is to explore the current state of research on cross-layer Fault Tolerance in underwater sensor networks and to identify any gaps in existing knowledge. Notably, there has been limited effort from the research community in studying Fault Tolerance in underwater sensor networks. To address this limitation, the used criteria (See Publication II) has been expanded to include papers that cover specific aspects of fault-tolerance. Additionally, the author also considers generic terrestrial fault tolerance in sensor networks, since research on faults in underwater sensor networks is scarce. The current chapter is based on Publication II.

2.2 Method

The methodology follows the PRISMA [105] guidelines for systematic reviews. In order to obtain a relevant sample in the field of fault-tolerant techniques in USNs, online environments were searched and results analyzed. A critical amount of papers was not reached using the initial under-water criteria, and the criteria were expanded to include also relevant non-marine-specific (terrestrial) papers. Many of the techniques, approaches, and tools developed for terrestrial networks can potentially be adapted for use in underwater sensor networks.

2.3 Results

To further emphasize the differences in research focus between marine and terrestrial sensor networks, the author presents a radar diagram in Figure 2. The diagram illustrates eight specific areas of interest.

Figure 2 shows that a considerable portion of marine research (depicted in blue) is devoted to underwater wireless communication. Some attention is also paid to underwater fault tolerance techniques, but there is minimal research on underwater cross-layer fault tolerance. Areas such as underwater energy efficiency and scalability receive more coverage compared to underwater vehicles (mobility) and security. On the other hand, papers related to terrestrial techniques (displayed in green) primarily focus on fault tolerance, including cross-layer fault tolerance, while energy efficiency and security aspects receive comparatively less attention.

The high research effort in marine wireless networking, as evident in Figure 2, confirms the claim by [78] that progress in the Internet of Underwater Things (IoUT) has been slow due to the unique challenges posed by underwater wireless sensor networks. Specifically, the main challenges for IoUT lie in the differences between underwater wireless sensor networks and terrestrial wireless sensor networks [78], for example, radio signals are absorbed in a short distance under water making them unusable, recharging devices may be complicated because physical access may be limited, water flow may move the devices, and pressure and corrosion may lead to flooding of the electronics.

2.3.1 Fault Tolerance Tasks

The objective of the current section is to define a taxonomy of Fault Tolerance tasks to help categorize the identified papers. The Fault Tolerance tasks are based on more general Fault Tolerance principles from Reference [147, 21]. Figure 3. shows the taxonomy of Fault Tolerance tasks applicable in USNs and how they affect each other. While the Design and initial Deployment of USNs contribute to Fault Prevention and Prediction abilities, Data Collecting techniques at the run-time contribute also to Fault Detection and Fault

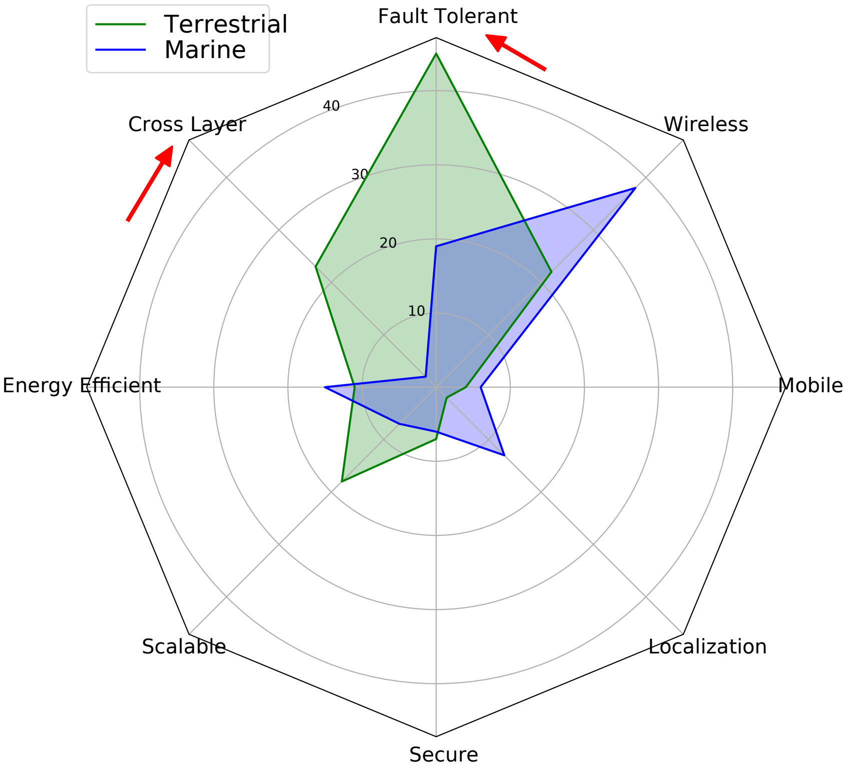


Figure 2: A radar chart of the analyzed papers addressing the main specific areas.

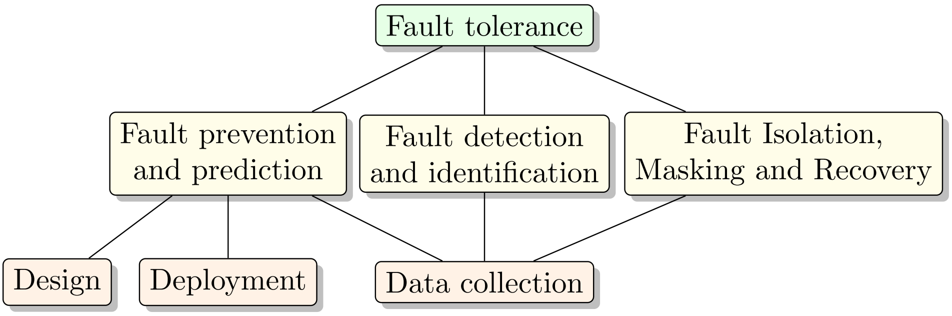

Fault tolerance
Fault prevention and prediction
Fault detection and identification
Fault Isolation, Masking and Recovery
Design
Deployment
Data collection

Figure 3: Taxonomy of Fault Tolerance tasks in USNs.

Recovery stages of the system.

The techniques under consideration can be categorized into the following groups:

- Fault Prediction and Prevention

  This task is about both preventing a fault to happen, as well as about proactive fault avoidance. Sensor networks can prevent certain faults from happening by design and/or deployment aspects. A disadvantage of fault prevention is a potentially increased system complexity. Fault avoidance, in turn, includes manufacturing testing and verification, which have a high cost often exceeding that of the entire design process.

- Fault Detection and Identification

  One of the central parts of Fault Tolerance is Fault Detection and Fault Identification of affected components which can, for instance, be performed by utilizing data collection with ping messages. Without Fault Identification, for instance, sensor node and network faults may be hard to distinguish. A disadvantage of Fault Detection and Fault Identification may be additional energy requirements and network congestion.

- Fault Isolation, Masking, and Recovery

  Isolation, masking, and recovery are different techniques for repairing a fault, minimizing the effect of a fault, or avoiding it to turn to system failure. Identified faults can be isolated, masked, and sensor network recovered, for instance, redirecting traffic through healthy backup components. Fault Recovery can ensure overall system operation even when components degrade. The downside may be the cost of adding components to ensure redundancy.

2.3.2 Comparative Analysis

All the papers that were included in the survey are listed in Table 1. The table includes information about the targeted extra-functional aspects and Fault Tolerance task(s). In addition, the Marine column in Table shows if the listed paper is explicitly touching aquatic environments. The papers are ordered by their order of citation within this survey paper. Papers that are not directly cited in the text but still listed in Table 1 are ordered chronologically by the publishing year. Papers that are not included in the analysis but are cited (e.g., definitions) have not been included in the table.

It can be seen from Table 1 that only two papers address both marine and cross-layer Fault Tolerance aspects. However, in the work targeting cross-layer analysis of error control [41], the term 'cross-layer' does not apply to the system stack but only to the communication protocol layers. Another work authored by the authors of this survey [155] is focusing on data-driven cross-layer Fault Tolerance. Thus, there is a serious gap in research addressing cross-layer Fault Tolerance in underwater sensor networks.

Regarding other extra-functional aspects, security in marine environments is addressed by six marine enrivonment related papers and is focusing on securing wireless communication [91, 61, 5], authentication [22], and hybrid attacks [60]. On scalability, seven marine-related papers were identified, and underwater scalability has been researched, for instance, in the context of monitoring underwater pipelines [104]. On Energy-efficiency, there were 14 marine-related papers identified, and extensive focus has been on energy-efficient underwater wireless protocols [43, 171, 167, 163, 159, 68, 124] and less on other aspects. Open research issues from all the mentioned extra-functional aspects are discussed in the following section.

Table 1: *Categorized papers.*

Pub.	**Year**	**Extra-functional aspect**				**Marine**	**Fault Tolerance (FT) Tasks and other Research Areas**
		Secure	Energy-efficient	Scalable	Cross-layer		
[78]	2017	-	-	-	-	+	FT design, survey, wireless
[3]	2019	-	-	-	-	-	FT detect/recover, survey
[88]	2018	-	-	+	-	-	FT detect/recover
[80]	2013	-	-	+	+	-	FT detect/recover, wireless
[116]	2007	-	-	-	+	-	FT detect/recover, survey, wireless
[39]	2009	-	-	-	-	+	FT detect, wireless
[41]	2012	-	-	-	+	+	FT detect/recover, wireless
[67]	2011	-	-	-	+	-	FT design/detect/recover
[53]	2013	-	-	-	+	-	FT design/detect/recover, vehicle
[97]	2012	-	-	-	-	-	FT prevent
[140]	2012	-	-	-	-	-	FT prevent
[127]	2016	-	-	-	+	-	FT prevent/detect/recover
[76]	2000	-	-	-	-	-	FT prevent/detect/recover
[143]	2012	-	-	-	-	-	routing protocol, survey , wireless
[144]	2018	-	+	-	-	+	sensor network, wireless
[152]	2013	+	+	+	-	-	survey, wireless, routing protocol
[113]	2016	-	-	-	-	+	routing protocol, sensor network, wireless
[74]	2018	-	-	-	-	+	survey, wireless, sensor network
[161]	2007	-	-	-	-	-	deployment, localization, sensor network, wireless
[71]	2004	-	-	-	-	-	deployment, sensor network, wireless
[42]	2020	-	+	-	-	+	sensor network, wireless, FT recover
[24]	2008	-	-	-	-	-	deployment, sensor network, wireless
[14]	2018	-	+	-	-	-	sensor network, wireless, FT detect
[55]	2017	-	-	-	-	+	wireless, sensor network
[56]	2017	-	-	-	-	+	wireless, sensor network, FT detection, FT recovery

Continued on next page

Table 1 – continued from previous page

Pub.	Year	Extra-functional aspect				Marine	Fault Tolerance (FT) Tasks and other Research Areas
		Secure	Energy-efficient	Scalable	Cross-layer		
[18]	2016	-	-	-	-	+	framework, wireless, sensor network
[118]	2015	-	-	-	-	+	framework wireless, sensor network
[119]	2013	-	-	-	-	+	framework, wireless, sensor network
[5]	2015	+	-	-	-	+	framework, wireless, sensor network
[21]	2010	-	-	-	+	-	FT design
[95]	2013	+	+	-	+	-	wireless, sensor network
[58]	2011	-	-	+	-	-	FT design
[110]	1995	-	-	-	-	-	sensor network, FT detect, FT recover
[112]	1992	-	-	-	-	-	FT design
[12]	2014	-	-	-	-	-	sensor network
[25]	2012	-	-	-	-	-	sensor network, deployment
[29]	2016	-	-	-	+	-	sensor network, FT detect, FT recover
[50]	2015	-	-	-	-	-	survey, sensor network, wireless, FT detect
[2]	2019	-	-	-	-	+	vehicle, FT recovery
[26]	2011	+	-	+	-	-	FT detect, FT recover
[102]	2016	-	+	+	+	-	survey, wireless, FT detect, FT recover,
[137]	2007	-	-	-	+	-	sensor network, FT detect, FT recover
[34]	2010	-	+	+	+	-	FT detect, FT recover
[30]	2017	+	-	-	-	-	survey, sensor network, wireless
[154]	2017	-	-	-	+	-	survey, FT detect, FT recover
[103]	2010	-	-	-	+	-	FT detect, FT recover
[66]	2014	-	-	-	+	-	FT detect, FT recover
[155]	2020	+	-	+	+	+	sensor network, FT detect
[163]	2012	-	+	+	-	+	FT detect, FT recover, sensor network
[167]	2016	-	+	+	-	+	sensor network, routing protocol, survey, FT detect, FT recover

Continued on next page

Table 1 – continued from previous page

Pub.	Year	Extra-functional aspect				Marine	Fault Tolerance (FT) Tasks and other Research Areas
		Secure	Energy-efficient	Scalable	Cross-layer		
[91]	2016	+	-	+	-	+	wireless, sensor network
[31]	2017	-	-	+	-	+	localization, sensor network, FT recover
[104]	2011	-	-	+	-	+	sensor network, FT detect
[16]	2000	-	+	+	-	-	localization, sensor network
[111]	2001	-	-	-	-	-	FT prevent
[168]	2002	-	+	+	-	-	wireless, sensor network
[33]	2004	-	-	-	-	-	FT design, sensor network
[15]	2005	-	-	-	+	-	cross-layer, FT design, FT recover, framework, sensor network
[64]	2006	-	-	-	-	+	sensor network, wireless
[92]	2008	-	-	-	-	-	wireless, FT detect, sensor network
[160]	2008	-	-	-	-	+	sensor network
[81]	2009	-	+	+	-	-	wireless, FT design, sensor network
[150]	2009	-	-	-	-	+	localization, sensor network
[166]	2009	-	-	-	-	+	localization, wireless, sensor network
[6]	2010	-	-	-	-	-	sensor network, survey
[83]	2011	-	-	-	+	-	vehicle, FT detect, FT recover,
[146]	2011	-	-	-	-	-	vehicle, FT detect
[133]	2011	+	-	-	-	-	sensor network,
[68]	2011	-	+	-	-	+	wireless, sensor network, routing protocol
[117]	2011	+	-	-	-	-	sensor network
[164]	2011	-	-	-	-	+	wireless, sensor network, routing protocol, FT recovery
[40]	2012	-	-	-	-	+	sensor network
[41]	2012	-	-	-	-	+	wireless, sensor network, FT detection, FT recovery
[100]	2013	-	-	+	+	-	wireless, sensor network, FT detect, FT recover

Continued on next page

Table 1 – continued from previous page

Pub.	**Year**	**Extra-functional aspect**				**Marine**	**Fault Tolerance (FT) Tasks and other Research Areas**
		Secure	Energy-efficient	Scalable	Cross-layer		
[151]	2013	-	-	-	-	-	FT detect
[57]	2013	+	-	+	-	-	wireless, sensor network,
[59]	2013	-	-	+	-	+	localization, sensor network
[4]	2013	+	-	+	-	+	vehicle
[43]	2013	-	+	-	-	+	sensor network, wireless
[114]	2014	-	-	-	+	-	wireless, sensor network
[126]	2014	-	-	+	-	-	wireless, sensor network
[87]	2014	-	+	-	-	-	wireless, sensor network, routing protocol
[172]	2014	-	-	-	-	+	sensor network
[134]	2015	-	+	-	-	+	sensor network, wireless
[10]	2015	-	-	-	-	-	FT mask
[13]	2015	-	-	-	+	-	sensor network
[169]	2015	-	-	-	-	+	localization, wireless, sensor network
[62]	2015	-	-	-	-	+	localization, wireless, sensor network, deployment
[153]	2015	-	-	-	-	+	wireless, sensor network, routing protocol
[61]	2015	+	-	-	-	+	wireless, sensor network
[128]	2016	-	-	-	+	-	FT detect, FT recover,
[159]	2016	-	+	-	-	+	sensor network, wireless
[171]	2016	-	+	-	-	+	wireless, sensor network, routing protocol
[135]	2016	-	-	-	+	-	FT design, FT detect
[36]	2016	-	-	+	-	-	survey
[93]	2016	-	-	-	-	+	localization, vehicle
[94]	2016	-	-	-	-	+	sensor network, wireless, localization
[79]	2016	-	-	-	-	+	vehicle, sensor network
[22]	2016	+	-	-	-	+	sensor network, wireless
[84]	2017	-	-	-	+	-	FT detect

Continued on next page

Table 1 – continued from previous page

Pub.	**Year**	**Extra-functional aspect**				**Marine**	**Fault Tolerance (FT) Tasks and other Research Areas**
		Secure	Energy-efficient	Scalable	Cross-layer		
[106]	2017	-	+	+	-	-	sensor network, survey
[124]	2017	-	+	-	-	+	sensor network, routing protocol
[145]	2017	-	+	-	-	+	wireless, sensor network, routing protocol
[20]	2017	-	+	-	-	+	sensor network, wireless
[44]	2017	-	+	-	-	+	localization, wireless, sensor network
[77]	2017	-	-	-	-	+	survey, sensor network, wireless
[108]	2017	-	-	-	-	+	localization, wireless, sensor network
[141]	2017	-	-	-	-	+	vehicle
[32]	2017	-	-	-	-	+	localization, sensor network
[7]	2018	-	-	-	+	-	FT detect, FT recover
[136]	2018	-	-	-	-	+	clustering, sensor network, routing protocol, FT detect, FT recover
[28]	2018	-	-	-	-	+	sensor network, FT detect, FT recover
[148]	2018	-	-	-	-	+	wireless, sensor network, fault, FT detect, FT recover
[165]	2018	-	-	+	-	-	vehicle, sensor network, wireless
[63]	2018	-	-	-	-	+	localization, wireless, sensor network
[142]	2018	-	-	+	-	+	localization, sensor network
[19]	2020	-	-	-	-	-	sensor network, FT detect, FT recover
[60]	2020	+	-	+	-	+	sensor network, wireless
[35]	2020	-	-	-	-	-	sensor network, FT detect
[75]	2020	-	-	-	-	+	sensor network, wireless, routing protocol, vehicle, FT detect; FT recover
[120]	2020	-	+	-	-	+	wireless, sensor network, fault, FT recover, FT detect
[107]	2020	-	-	-	-	-	sensor network, wireless, FT detect, FT recover
[162]	2020	+	+	-	+	-	sensor network, survey. FT detect, FT recover
[73]	2021	+	+	+	-	+	sensor network, survey

Continued on next page

Table 1 – continued from previous page

Pub.	**Year**	**Extra-functional aspect**				**Marine**	**Fault Tolerance (FT) Tasks and other Research Areas**
		Secure	Energy-efficient	Scalable	Cross-layer		
[121]	2022	-	+	-	-	-	sensor network, routing protocol, wireless
[11]	2023	+	+	+	+	-	wireless, sensor netwok, data aggregation, FT detect
[173]	2023	+	+	+	-	+	survey

2.3.3 Open Research Issues

The open research issues identified are presented according to the categories of extra-functional aspects reported in Table 1.

Security and faults are interrelated concepts [26] as faults can enable new intrusion vectors [34] and security intrusion can lead to faults. **Energy** usage need to be addressed as USN-s may not have unlimited power supply and cross-layer approach is considered more energy-efficient [21] than single layer. **Scaling** benefit is to allow lower cost per functionality [34]. At the same time large scale fault tolerant systems are researched without paying special attention to energy and communication constraints [26]. Current thesis addresses these issues to some extent from the cross-layer perspective (see for example Section 3.3).

2.4 Chapter Summary

The current chapter presented fault tolerant techniques in USNs and pointed out open research issues in this field. The techniques were divided into groups according to the new taxonomy of Fault Tolerance tasks, and papers utilizing these techniques were shown in sections corresponding to the tasks.

The conducted survey was the first to investigate the state-of-the-art in Fault Tolerance, particularly cross-layer Fault Tolerance, in USNs. According to the survey, there is a lack of research about the cross-layer Fault Tolerance aspect for underwater sensor networks, which is covered in this thesis.

3 Preliminaries: Sensors, Sensor Network Framework, Data Layers, and Case Study Scenarios

3.1 Motivation

This chapter further elaborates on the devices, framework and concept that are not exhaustively presented in published papers; additional information is provided here to provide a better understanding of these essential elements that are used in the presented works. In Section 3.2 the author is describing the sensors used in Chapters 4-6. These sensors were not built by the author of this thesis. In Section 3.3 the author describes the software framework developed and used exquisitely for the current thesis Chapters 5, 6 by the author. In Section 3.4 the implementation of data layers proposed by the author is explained, which was implemented in the software framework and used in Chapters 5,6.

3.2 Sensors Used for Experiments

In the next sections, we cover sensor networks for two case studies - an underwater flow monitoring and an indoor climate sensing.

3.2.1 Underwater Sensor Network Installation

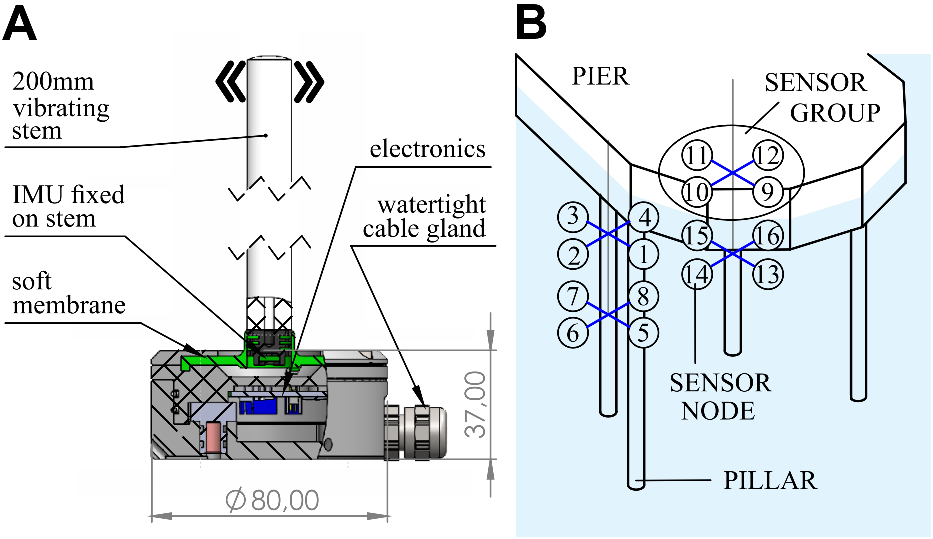


Figure 4: A) Sensor Node B) Harbor Installation

The proposed approach was evaluated in an underwater network to monitor the sea currents in the harbor. Sensor nodes are installed on the underwater harbor infrastructure to notify approaching ships about the water flow around the piers. The goal is that berthing ships get information about the flow and turbulence from Sensor Nodes installed on the pillars of the pier. The Sensor Nodes are connected to the Sensor Gateway with underwater cables over RS-485 serial communication, and thus the configuration of the underwater sensor network is fixed.

Hydromasts [130, 132, 125, 131] are sensors that have previously been used for underwater flow sensing [130, 132, 131], hydromorphological classification [130], and surface vessel localization [125]. In this thesis Hydromasts are used for flow sensing and are from here on referred to as Sensor Nodes.

The Sensor Node applied in the case study is shown in Figure 4. A and its detailed description is given in [132]. The Sensor Nodes measure the flow magnitude and direction from the stem vibrating in the flow. An IMU (Inertial Measurement Unit) embedded into the Sensor Node calculates the accelerations of the stem in x and y directions. In the current implementation, to estimate the flow magnitude from the IMU data, a 15 s series of IMU Raw data is transformed with FFT (Fast Fourier Transform) into a frequency domain in 120 s intervals, and the PSD (Power Spectral Density) is used to find flow direction and magnitude using calibrations described in [132]. All the Sensor Nodes are wired to the Sensor Gateway in a star network topology.

The Sensor Nodes are installed around the pillars at two different depths so that at both depths 4 Sensor Nodes are attached around the pillar at 90 degrees angle from each other forming a logical Sensor Group. This is necessary because, depending on the flow direction, the pillar itself always shelters some of the Sensors Nodes from the flow. Therefore, each aggregation of 4 Sensor Nodes (Sensor Group) is used to estimate the flow at each point. Note, however, that the values of the Sensor Nodes in the same Sensor Group are correlated. Knowing how the flow should behave around obstacles helps us identify failing Sensor Nodes as well as to estimate how the readings of the sensors should be correlated.

In total, the installation has 16 Sensor Nodes, grouped into 4 aggregations (Sensor Groups) of 4 sensors as shown in Figure 4.B.

3.2.2 Indoor Sensor Network Installation

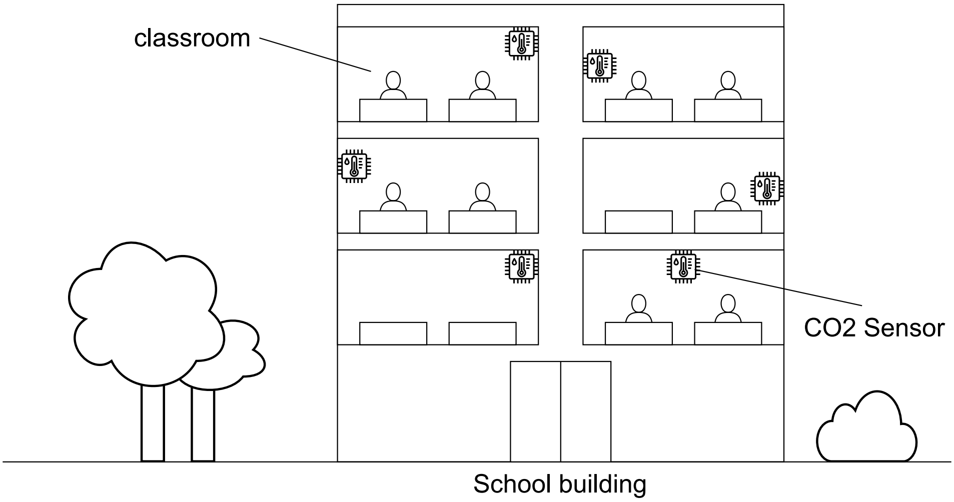


Figure 5: School building installation

For the indoor climate sensing experiment covered in Section 4 the sensors used were as follows. Commercial indoor air quality sensors manufactured and designed by Smart Temp Australia Pty Ltd model number SMT-IAQ3 [149] were used. The producer claims that the accuracy of the temperature sensor is +/- 0.5 ° C at 25 ° C (operating range −5 to 50 ° C) and the accuracy of the relative humidity (RH) sensor is +/- 5% at 25 ° C and 30% to 80% RH (range 0 to 95 % RH). The CO_2 sensor is a non-dispersive infrared sensor with

+/- 30ppm accuracy at 25°C, and the operating range is 0-2000 ppm. The sensor applies auto-calibration as the manual [149] states that – "*The CO_2 sensor within the SMT-IAQ3 has an advanced learning self-calibrating function. This calibration process takes place over an 8-day period.*". The sensors were installed into 56 different rooms in a well-ventilated school building in Estonia and the data logged during 6 month time period (see Figure 5).

The installation of the sensor network for the indoor experiment was implemented by a commercial third party and the framework described in 3.3 was not used. The author of the current thesis analyzed and developed an algorithm based on the already gathered sensor data (see Chapter 4).

3.3 Implemented Software framework for sensor networks

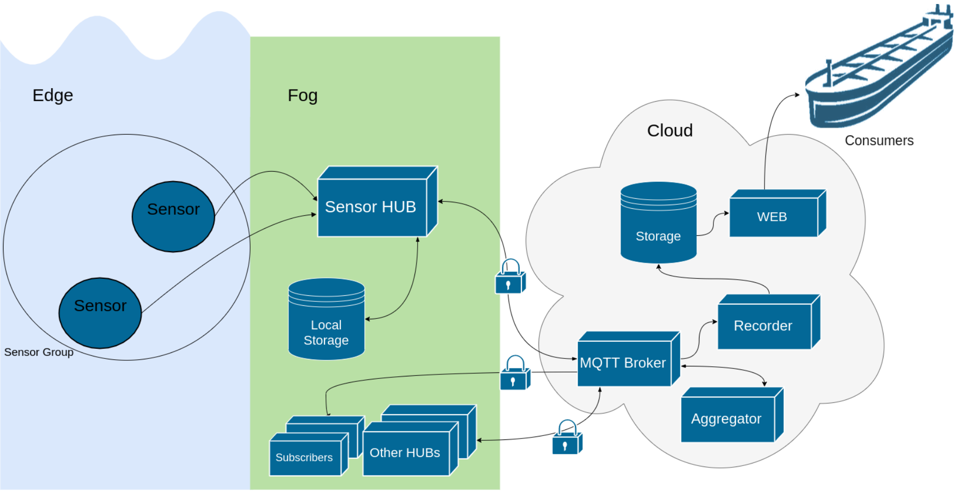


Figure 6: Server-based Underwater Sensor Network framework software architecture.

This subsection presents the software framework for underwater sensor networks implemented in the current thesis. The implemented software framework is also presented in the publication V. The author provides an overview of related works, describes the developed components, and gives an overview of the high-level software architecture of the framework.

3.3.1 Overview of Related Works

[86] has proposed a general Sensor Network application approach. The threat of error propagation is addressed. It covers Input, Signal processing, and Output. Data processing and prediction are also mentioned. However, no clear methodology choices are proposed, the paper is very general, and no implementation details are covered.

[99] is introducing the Fault Tolerance Event Manager (FaTEMa) into IoT systems. It is a multi-layer approach and divides challenges into Heterogeneity, Adaptability and Evolution, Distributed Decision Mechanism, and Information Redundancy. Resource consumption was considered and scalability was addressed using distributed decisions. Security is mentioned, but no details like Infrastructure as Code and ease of development and rapid deployment are addressed.

Other works [96, 138, 17, 38, 89, 115, 1, 90, 65] lack the following advantages that are implemented in the sensor network framework developed by the author of the current thesis and presented in the publication V.

- Infrastructure as Code (IaC)
  - Scalability, immutable infrastructure, ease of operational maintenance. Reduce time and resource-consuming software configuration and maintenance. Automatic scalability from cloud reliance and easy infrastructure multiplication for new installations. Immutable infrastructure makes it possible to maintain more complex systems with the same resources.
- Containerized edge application.
  - Platform independence, configuration automation
- Edge cloud VPN.
  - Fault management, edge devices online monitoring, IoT fleet management. Allows monitoring state of devices near real-time, which is not possible without IP network (e.g. LoRa). Proactive fault management is possible.
- Transparent remote release process.
  - Rapid bugfixes, reduced downtime. Not possible without IP network (e.g. LoRa). Software code will be accessible, and new releases do not need physical access to sensors.
- Security
  - Every Sensor Hub has own SSL public/private key pair signed by public CA, authentication is done via private keys, validated on cloud side and whole traffic travelling over public network is encrypted into SSL tunnel.
  - Multitenant
  - Open Source

3.3.2 High-Level Software Architecture of the Framework

For the purpose of managing underwater sensor networks and their faults, the author developed and implemented an initial software framework for running the components on one or multiple network servers. The initial implementation of the framework architecture is shown in Figure 6. In the following, a brief overview of the developed components divided into logical Edge, Fog, and Cloud [8] computing environments is provided.

1. The Edge computing environment includes *Sensor Nodes* that measure environmental data and transfer it to a Sensor HUB.
2. The Fog computing environment consists of a *Sensor HUB* that is connected to multiple sensor nodes that can form *Sensor Groups*. A Sensor Hub gathers sensor data, processes it, and publishes data further into the Broker in the Cloud Layer.
3. The cloud computing environment has a *Broker* component, which acts as a communication bus between the Cloud and Fog layers, as well as between functional components in the Cloud layer. The term Broker originates from the MQTT [72] standard; however, its functionality is not vendor-locked. The Cloud layer can also include an *Aggregator* that manipulates the data according to defined domain-specific rules and publishes the manipulated (aggregated) data back to the Broker.

> The *Recorder* is subscribed to the Broker and listens to the incoming data. It does not manipulate the data in any way and only records them using predefined rules. *Subscribers* can subscribe to the Broker to the data interested. This allows accessing different data feeds that pass the Broker with small latency. The *Web component* is presenting data from the Storage to End Users. The name is an abstraction that does not have to be a Web application but can be any kind of user interface.

A more detailed view of the software framework using a cloud service provider is presented in Figure 7. Components are divided by the used environment into 4 layers. Edge layer (red) contains components with limited network connectivity, computational and/or storage resources. There are 2 cloud layers used. Balena cloud (blue) is used for edge components monitoring and management. Amazon Web Services (orange) contains main cloud components for data collection, additional processing, and presentation. The external access layer (white) is for 3rd-party integrations, software deployment, fleet management, and data access.

Components in Figure 7 presented with blue background are administered services provided by Balena, orange background means administered services managed by AWS and green background notes custom developed components. The continuous lines in Figure 7 show network connections established by the system, dashed lines human-activated and on-demand connections.

For AWS services, IaC (infrastructure as code) is used, specifically AWS Cloudformation CDK. The source code for self-developed components as well as IaC is kept in GIT repositories. Self-developed components are using Python 3.9 and ReactJS (for the web interface).

On the Edge layer, there are sensor hub devices. The sensor hub hardware is Raspberry Pi 3, Beaglebone Black/Green, or other single board computer (SBC). It can store the data of each sensor locally on an SD card or USB memory stick, and send data via MQTT or LoRaWAN to AWS IoT Core. Hydromast sensors that can be reached with a cable, can connect to a single sensor hub. Usually, 1-5 sensors connect to one sensor hub. The sensor hubs run on Taltech developed Sensor Hub application (see Figure 7) on BalenaOS. BalenaOS is an operating system that is based on Linux kernel and is optimized to run Docker containers on embedded devices. Sensor hubs are managed through BalenaCloud or OpenBalena servers. These servers keep all the devices in groups called fleets, which enables monitoring, developing, configuring, and updating the software on specific or all connected fleet devices simultaneously. A Sensor Hub can be either offline or online. In case of being online, a CloudLink OpenVPN connection is established to Balena cloud services and (depending on configuration) to MQTT Broker (see Figure 7). Balena cloud VPN connection allows to monitor the health of the device, change the configuration variables of the hub, and manage software releases on it. MQTT connection allows to upload data and respond to commands.

3.4 Proposed Layered Data Model

In this section, we look at the sensor network from the perspective of the sensor data. Table 2 presents the data-driven layers of the sensor network architecture proposed by the author of this thesis. There are three data layers - Raw, Processed and Aggregated that correspond to different functionalities that can be loosely connected to given computing environments as shown in the Table.

In the *Raw data* layer (i.e. layer L_1), there are data that are measured and transferred by the set of Sensor Nodes $S = \{s_i\}$. The Raw data layer holds Sensor Node data before

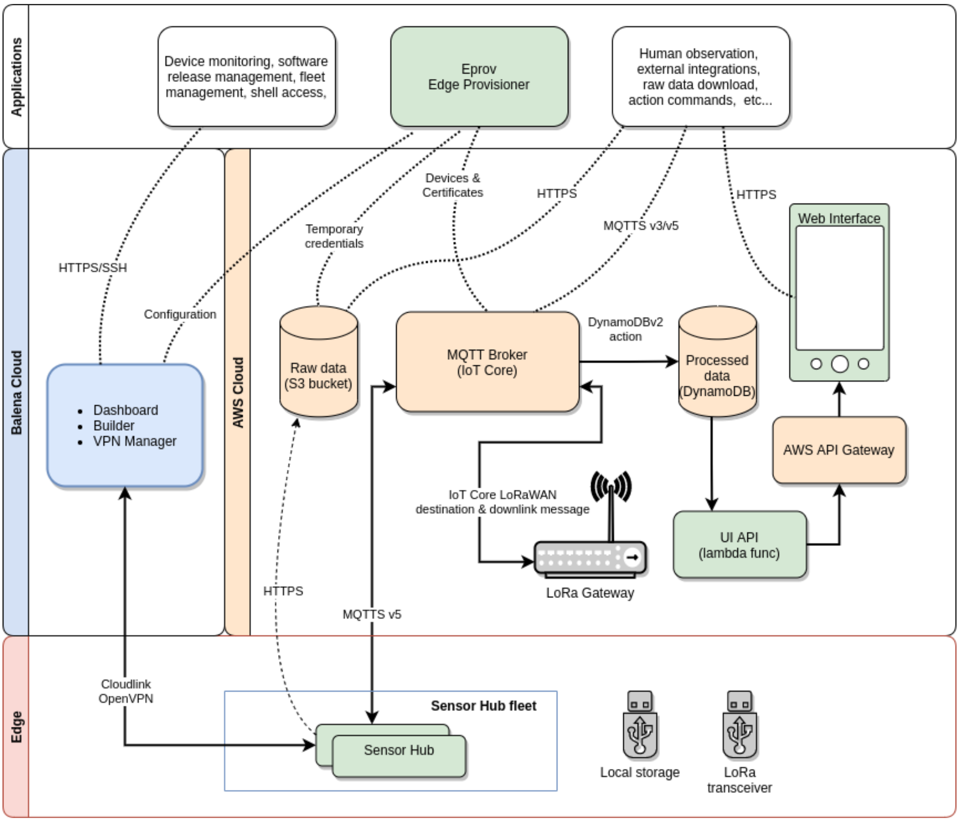


Figure 7: The implemented underwater framework architecture.

transformations and calibrations.

Processed data (layer L_2) is generated by the Sensor Gateway (s) (see Section 3.3) from Raw data. The processing step is essential in the current application (see Section 3.2), where sensor acceleration has to be transformed into the frequency domain. Additionally, during the processing outlier data is smoothed out and it is determined whether the data are within the measurement limits.

Aggregated data (layer L_3) is generated for the Sensor Groups $G = \{g_j\}$ by the Aggregator component (see Section 3.3) from the Processed data of the Sensor Nodes $\{s_i\}$. The purpose of the Aggregating process is to provide transition from single Sensor Nodes' data to Sensor Groups' data.

Table 2: Data layers and their mapping to computing environments

Data Layer	**State**	**Computing environment**
L_1: Raw	direct data from sensors	Edge
L_2: Processed	pre-processed data from sensors	Edge/Fog
L_3: Aggregated	data combined to sensor groups	Cloud

More formally, the data processing and aggregation takes place as follows. Let us assume that sensors measure and send values periodically and the event of receiving a valid test packet at the Raw data layer L_1 by sensor s_i at a discrete time instance t be $\rho(i,t)$. All the valid $\rho(i,t)$ measurement values received during a time interval T, containing n time instances, are pushed into the Raw data queue of length n, which acts as a First-In-First-Out (FIFO) data buffer:

$$Q = \{q_1, q_2, ..., q_n\},$$

where $q_1, ..., q_n$ represent n latest valid $\rho(i,t)$ measurement values.

To obtain the processed data at layer L_2, the data in Q are processed by a signal processing function Ψ to obtain the processed data value of $\sigma(i,t)$, i.e.

$$\sigma(i,t) = \Psi(Q).$$

Subsequently, the processed data value $\sigma(i,t)$ of the node s_i is passed through a calibration function Φ_i, i.e. the calibrated processed value of sensor s_i at time t is

$$\gamma(i,t) = \Phi_i(\sigma(i,t)).$$

Subsequently, the aggregated value $\alpha(j,t)$ of the sensor group g_j in layer L_3 is calculated by an aggregation function Ω on calibrated values $\gamma(i,t)$ of the sensor nodes s_i that belong to the sensor group g_j, i.e.

$$\alpha(j,t) = \Omega(\{\gamma(i,t) | s_i \in g_j\}).$$

Finally, the aggregated value $\alpha(j,t)$ is converted to the physical quantity $\beta(j,t)$ required by the end user by applying the conversion function Ξ:

$$\beta(j,t) = \Xi(\alpha(j,t)).$$

3.5 Chapter Summary

In the current chapter, the author covered the pereliminaries of the Data-driven cross-layer Fault management in underwater sensor networks. In the next chapter, we proceed with the proposed fault management framework by presenting the data-driven method for sensor data processing.

4 Data-Driven Method for Processing Raw Sensor Data

4.1 Motivation

Often direct environmental measurements of sensor networks need data processing to improve data quality. The purpose of this processing can be mitigating hardware faults but possibly also other misbehavior of the specific devices, or for instance the cases where not absolute but relative values are needed.

The Data-Driven Fault-Resilient Cross-Layer methods proposed in this thesis are general and not dependent on the used environment. However, underwater conditions are much harsher and faults more frequent. Although this thesis is mainly focused on underwater sensor networks, in this chapter, that is based on Publication III, due to the lack of the sufficient underwater data, a terrestrial sensor network of in-door climate is considered and analyzed. The applicability of the methods to UWSN-s is covered in more detail in Section 4.2.

In-door climate sensors can measure temperature, humidity, and CO_2 concentration in the rooms. These data are often used for ventilation control, air quality assessment, and occupancy detection. The measurement of the concentration of gases is technically complicated and may lead to significant measurement errors [109]. Additionally, the in-door sensors may have the auto-calibration function that shifts the zero level so that the measurements would not drift off. However, this creates jumps in the data and sometimes values below the outdoor CO_2 level. If the data is further used, occupancy is detected differently, ventilation would not function as designed, and the assessment would result in a different air quality class estimation.

Data-driven methods for processing of Raw data from CO_2 sensors were developed to improve the quality of ventilation control and in-door air quality assessment in the publication III on which the current chapter is based. Methods were tested and verified on data collected from 56 CO_2 sensors from a school building with balanced heat recovery ventilation in Estonia. The developed methods can be implemented in existing building management and In-door Air Quality analysis tools with reasonable computational cost.

Section 4.2 discusses the data correction method applicability in the context of Underwater Sensor Networks. Section 4.3 describes the development of the method. The results are described in Section 4.4 and Section 4.5 concludes the chapter.

4.2 Applicability in the context of USN's

Although this thesis is mainly focusing on underwater sensor networks, the current chapter discusses methods in an in-door sensor network. This section discusses the applicability of the described methods in the underwater context.

Identified underwater challenges (see Section 2) can be divided by the cause into environmental conditions and engineering constraints. The applicability of the techniques discussed in the current chapter by this division is the following.

- The underwater harsh environmental conditions like high pressure, high turbulence, and aquatic lifeforms may contribute to more frequent and different types of faults. While the cause of the faults may be different, the data behaviour in Raw, Processed, and Aggregated data layers, like redundancy and calibration errors, false readings, and decoding exceptions, is similar. Thus, the underwater environmental conditions allow to use the methods described in the current chapter.

- Additional optimization of methods may be necessary based on bandwidth and energy constraints. The processes may need to be redesigned to determine the appropriate IoT layer for their execution. However, the data correction methods are applicable also in USN's.

The technique presented in the current chapter aligns with the proposed data layer model (see Figure 1) by manipulating the Raw data to contribute to the Processed data layer. While the operative scenarios and fault models may be different in in-door and underwater environments, there is no distinction in data life cycle and behaviour on specific faults. It should be clarified that the example of the concrete sensor correction algorithm described in the current chapter, is aimed at the specific anomalies caused by the shifting of the sensors, either in in-door, underwater, or other environments. The method presented in the current chapter can be viewed as an optional pre-processing step for fault management discussed in the subsequent chapters.

4.3 Data Collection: Smart Building Case Study

In-door climate data was collected from a high school building for over 6 month period. The following typical cases of deviation from the expected behavior of the CO_2 measurements were identified and further analyzed:

- ***Outliers*** were identified as sudden change of the logged CO_2 concentration, which was followed by a sudden change of the same magnitude in the opposite direction.

- ***Incorrect baseline*** was identified as a significant difference of CO_2 concentration from the outdoor concentration of 400 ppm.

- ***Auto-calibration*** of the indoor climate sensor controller was identified as a sudden change in the logged CO_2 concentration to approximately 400 ppm with no change in the opposite direction thereafter.

- ***Potentially inadequate placement of CO_2 sensor*** was identified as sudden changes in the CO_2 concentration at reoccurring times of workdays when the ventilation system is either turned on or off. This could be caused by the non-uniform distribution of CO_2 in the room air caused by the air distribution solution.

- ***CO_2 concentration's dependency on the air temperature*** was identified as an unexplainable fluctuation of CO_2 concentration fluctuation during unoccupied hours that had a negative correlation with the air temperature fluctuation.

To help mitigating the above-mentioned problems, several on-line (sequential) and offline (retrospective) change point detection [85] and data smoothing algorithms were investigated. Change point detection helps identifying rapid variations in time series data. Accordingly, data smoothing can be used to eliminate peaks and noise caused by measurement errors. The author developed a low quantile method [70] that uses specified fraction of lower valued measurements to determine the current baseline. The algorithm is described in the following pseudocode block.

Algorithm: Return input vector with corrected baseline

```
function get_fixed_baseline(vec, window_size, bottom=1, base)
    result $\leftarrow$ []
    for $j \leftarrow 0$ to len(vec) $-$ window_size do
        $lp \leftarrow$ np.percentile(vec$[j : j +$ window_size$]$, bottom$)$
        result.append$(median(lp))$
    end for
    lb $\leftarrow$ np.concatenate$(($np.ones(window_size) $\times$ result[0], np.array(result)$),)$
    return vec $-$ lb $+$ base
end function
```

The algorithm to return the measurement vector with corrected baseline values is shown as an offline version in the pseudocode. The input variables are as follows : vec is the sensor measurement vector of floating point numbers, $window\_size$ is an integer with the size of the sliding window buffer (120 in case of 5 min measurement interval and 10h window), $bottom$ is an integer with the needed bottom percentile value and $base$ is the floating point actual baseline value (400 in our case, provided by legislation). The called functions are as follows: $np.percentile$ returns the requested percentile vector from the input vector, $np.concatenate$ joins multiple arrays and is used for the first values when the correct baseline cannot be determined yet.

The developed low quantile method is discussed in detail in the publication III. The method can detect and compensate the above-mentioned baseline and auto-calibration deviations, as well as also eliminate high outlier data. The time complexity of the method in the big O notation is $O(n)$ where n is the size of the data in the defined sliding time window (10 h for the classroom data used in the experiments). Being online as well as computationally light-weight enables its use also in real-time applications.

4.4 CO_2 Concentration Correction Results

The performance of the CO_2 base value correction algorithm used was qualitatively evaluated by comparing initial logged and corrected CO_2 level concentrations. Figure 8 illustrates the respective CO_2 concentrations during 5 weeks in all the classrooms. The initially logged CO_2 concentration outside expected occupied hours varied constantly over 100 ppm and was typically above 400 ppm. This highlights the need for further post-processing of logged data of the installed sensors despite the in-built auto-calibration function. The corrected values reflect the expected behavior. However, there were still some outliers with values significantly below 400 ppm due to the chosen quantile method, which should be removed by further developing the data-processing methods before applied for IAQ assessment or ventilation control.

The performance of the base-level correction method was additionally assessed by comparing the duration curves of initially logged and corrected data. The results of all the rooms with CO_2 sensors are given in Figure 9. Visually, the largest difference appears during periods when the CO_2 values are near outdoor level.

Table 3 shows the impact of the baseline level correction by comparing the logged values against different CO_2 concentration levels. The COVID-19 pandemic situation recommended ventilation control setpoint 550 ppm [129] for was exceeded during 89.1 and 74. 4% of the occupied period with initially logged and corrected data, respectively. Therefore, the impact of baseline-level correction would have been significant if the school had demand-controlled ventilation and the recommendations had been followed. The dif-

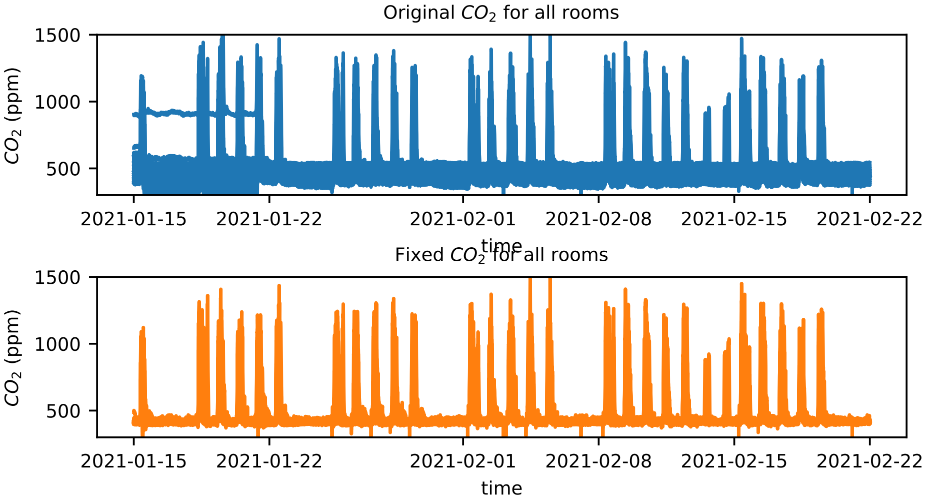


Figure 8: Example of CO_2 concentration of all rooms during 5 weeks with initially logged (above) and corrected (below) baseline values

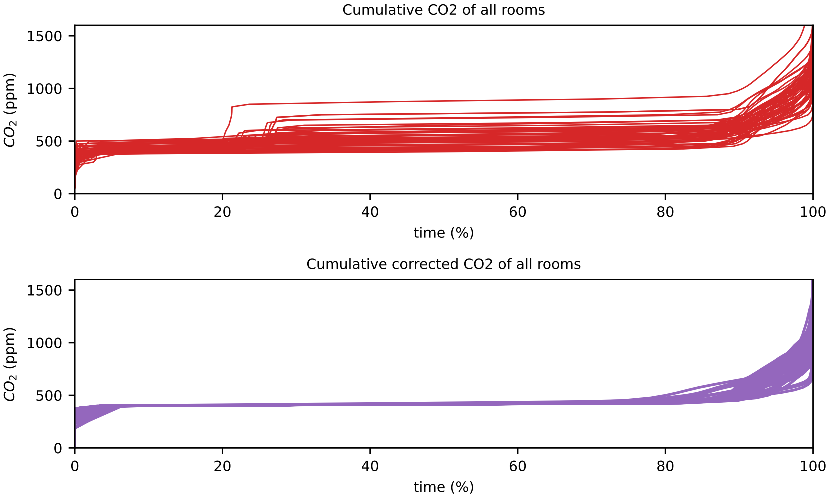


Figure 9: Cumulative CO_2 of all rooms in the same timeframe as previous figure

ferences in exceedances of 800 and 1000 ppm were even more significant. As initially logged CO_2 concentrations were generally higher than the corrected values, then the initially logged values would have prompted unnecessary disturbances in school work due to opening windows during classes or even stopping schoolwork due to increased infection-risk.

Table 3: The impact of baseline level correction of comparing logged values against different CO_2 concentration levels

Time of occupied period above the respective CO_2 level, %			
	550 ppm	800 ppm	1000 ppm
Initial	89.1	46.2	18.2
Corrected	74.4	27.7	5.9

4.5 Chapter Summary

In this chapter, a terrestrial sensor network for in-door climate monitoring was considered. Data-Driven Fault-Resilient Cross-Layer methods are general and not dependent on the used environment, but underwater conditions are much harsher and faults more frequent. Although this thesis is mainly focused on underwater sensor networks, in this chapter, that is based on Publication III, a terrestrial sensor network of in-door climate is considered and analysed, due to the lack of the sufficient underwater data. Data-Driven methods are not environment dependent and can be used for underwater as well as terrestrial sensor networks. The data from sensor networks are frequently used for assessment, automation, or as input to other processes. However, often when sensor data are flawed, it needs processing before moving on. In the current chapter, a method for CO_2 base level correction was developed and method's performance qualitatively assessed.

The method used percentile values from a sliding time window to identify the base level, and subsequently the data were corrected so that the base level would be constant. The best compromise between accuracy and delay in data correction was reached using 1% percentile values in a 10 hour sliding time window. The impact of the algorithm was significant when comparing the initially logged and corrected values against CO_2 concentration thresholds.

The method was developed to correct CO_2 concentrations in near real-time without knowing future values. Therefore, the current implementation is more effective in capturing the downward jumps of the CO_2 base level. The baseline correction method can be implemented also for regular In-Door Air Quality assessment, e.g. with weekly or monthly frequency, and for such application the method should be developed further to also effectively capture the upward jumps, which is currently delayed.

The developed method needs to be further tested based on data from inadequately ventilated buildings, building with different occupancy patterns, and sensors that might provide worse quality data that contain more outliers. Additionally, the post-processed CO_2 concentration needs comparison with more detailed measurements with calibrated sensors for further validation and development. Finally, the current implementation of the algorithm effectively captures downward shifts in the logged data, but not upward shifts. That behavior is caused by the minimum values of the sliding time window that cannot be easily predicted because of the online (i.e. near-real-time) nature of the proposed algorithm. Future development is needed to be able to capture upward shifts with a lesser delay.

5 Cross-Layer Fault Management in Data Processing

5.1 Motivation

Cross-layer resilience is the cooperation of multiple techniques from different layers of the system stack to achieve cost-effective error resilience, as a possible solution to design low-cost resilient systems [23].

Data-driven signal processing-based fault detection and diagnosis employ signal processing functions to identify faults and anomalies [37, 27, 52, 170, 54].

The current chapter is based on the publication I and combines two concepts mentioned above, introducing a data-driven, cross-layer fault-resilient architecture for sensor networks. The proposed approach does not focus on the system stack but rather on the data health of the sensor network.

The main purpose of a sensor network is to measure, gather, and deliver valid data. Although the root cause of invalid data lies somewhere in the system stack, the concept of data-driven cross-layer resilience is based mainly on the purpose of the sensor network and less on the implementation details. In the approach proposed in the current thesis, the data layers are part of the sensor network's essential functionality itself, and no additional dedicated hardware is needed. In proposed architecture data are collected exclusively by the functional sensors (the Raw data layer) that are part of the system. Further data layers include processed data and aggregated data from different sensor sources. As it is shown in this chapter, the approach allows detecting faults on a higher data layers even if they escape lower layer detection, and provides online diagnosis capabilities to the network in order to pin point the root cause and location of the error.

5.2 Fault Detection and Diagnosis at Different Data Layers

Figure 10 presents a flowchart of Raw and Processed data layer processes. When a serial packet arrives from a sensor node, it is decoded, and the sensor values are pushed to a fixed size raw data FIFO queue. When the FIFO queue is full, then the oldest values are disposed. Simultaneously, at the predefined time intervals, data from the raw data FIFO queue are cloned taking a still snapshot of the contents of the Raw queue. This allows to compute the Processed data layer values without interfering with the continuous Raw data layer update. For a more detailed definition, implementation, and notation of data layers, please see Section 3.4) or Publication I on which the current chapter is based.

Fault detection at Raw Data Layer L_1: The fault detection at the Raw data layer is executed when decoding the incoming sensor measurements (see Figure 10). Invalid packets at sensor nodes s_i (e.g., malformed or missing packets) at time instance t are filtered at this layer. In addition, the ratio of good to bad packets is accumulated. At this level, problems that may occur and cause invalid packets are as follows:

- Data Decoding Error - the data packet does not correspond to the correct length or byte values and cannot be read from current stream.

- Redundancy check failing - One of the values in the data packet is the name of the Sensor Node, which should stay constant and never change.

The amount of deterioration of a sensor on the Raw data layer is characterized by the difference on invalid and valid packets.

Fault detection at Processed data Layer L_2: Processed data layer consists of single sensor s_i values identified by index i values, similar to Raw Data Layer. This layer includes an incoming FIFO queue for the data from the Raw data layer of which a snapshot is taken

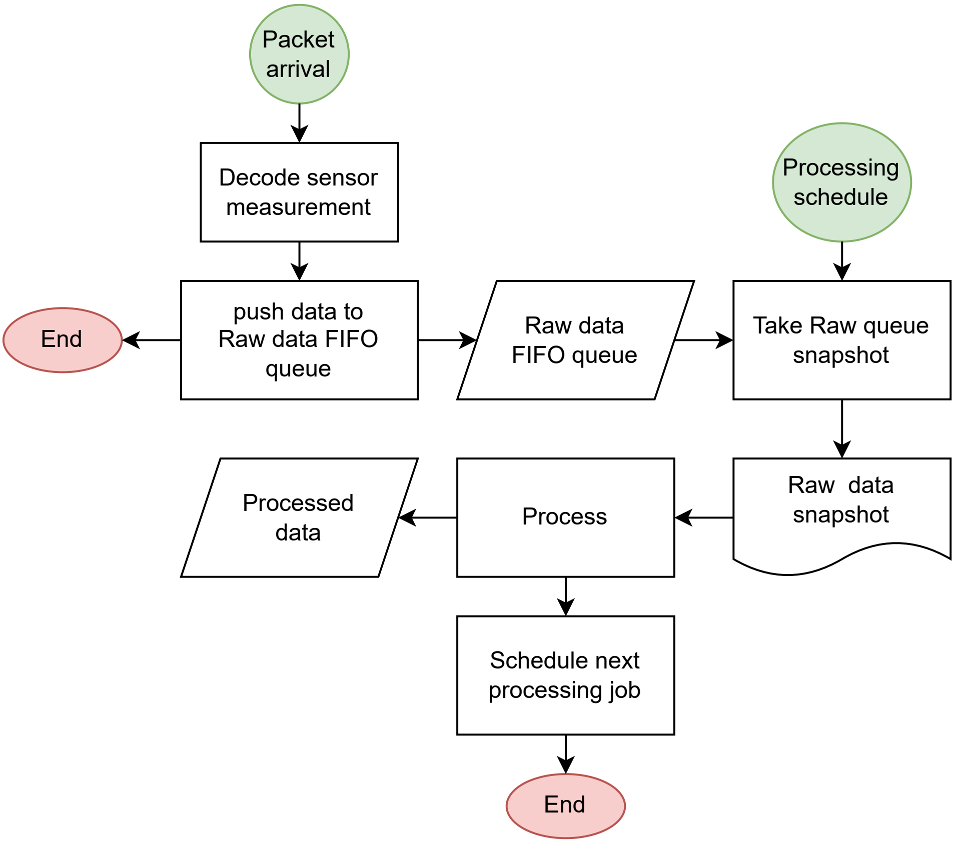


Figure 10: Raw and Processed data flowchart

at a predefined interval (see Figure 10). The contents of the queue are processed, and the results are assigned to the processed data collection that is then sent upstream. To be able to obtain the processed value (e.g. Fourier transorm), the Raw data FIFO queue has to be full. The processing steps at this data layer include the following.

- Not enough data - The FIFO queue is cleared after every data acquisition by the processed data layer. If not enough good packets from the Raw data layer came in during the last acquisition interval, the queue may not be full. This indicates that processing cannot be done.
- Filtering out data values outside the measurement boundaries.

The fault on the Processed data layer is detected when a processed value is outside predefined boundaries.

Fault detection at Aggregated Data Layer L_3: Single sensors s_i form a sensor group g_j identified by group index j. At Aggregated data layer, the sensor groups are managed, single Sensor Nodes' data and Health-maps are analyzed, combined, and aggregated. When a sensor fails, it is removed from its Sensor Group. If a critical number of sensor nodes have decayed then the group data cannot be trusted anymore and the sensor group is regarded as faulty. The aggregating methods are discussed more in detail in the next Chapter 6 and the aggregation processes flowchart presented on Figure 16.

5.3 Results

In current section the author presents the results of synthetic scenarios based on real data showing the benefits provided by the cross-layer data-driven fault management approach. In addition, some real-world examples of naturally occurred incidents are given, where faults were detected and diagnosed by the approach.

5.3.1 Graceful Data Degradation Enabled by Fault Management

In current subsection, the author is using real data recorded from natural measurements during a selected time interval. The faults are injected into these data by overriding data values in specific sensors by constant zero vectors. Thus, synthetic fault scenarios are created. This set of data values is then simulated to obtain the aggregated values for the sensor group.

Figure 11 shows the data degradation in synthetic scenarios based on real data, where the sensors were degraded. The comparison (see Figure 11) includes proportional differences of the aggregated data calculated using failing sensors compared to the good aggregated data scenario with operational sensors. Proportional differences are found using $max(x,y)/min(x,y)$ where x is the best scenario measurement value and y is correspondingly some other scenario measurement value at the same time. The orange line represents the case where one faulty sensor has been removed from the aggregation of the group value. This is equal to the good aggregated value. The blue line represents the case where one sensor is faulty (having $\sigma.dfq = 0$ where $\sigma.dfq$ denotes the dominant frequency value from the result of the data processing function) but is still included in the aggregated data computation. The red line represents the scenario where 2 sensors are faulty (with $\sigma.dfq = 0$). Finally, the green line shows the scenario where the 2 faulting sensors are removed from aggregated data computation by the fault manager. The experiments show that data-driven cross-layer fault management allows on average to improve sensor group measurement accuracy by 35% (1.0 × versus 1.35 × deviation from the good value) in case of single sensor errors and nearly twofold (1.72 versus 3.18 × deviation) in case of double sensor errors (see the average values for different scenarios in Figure 11).

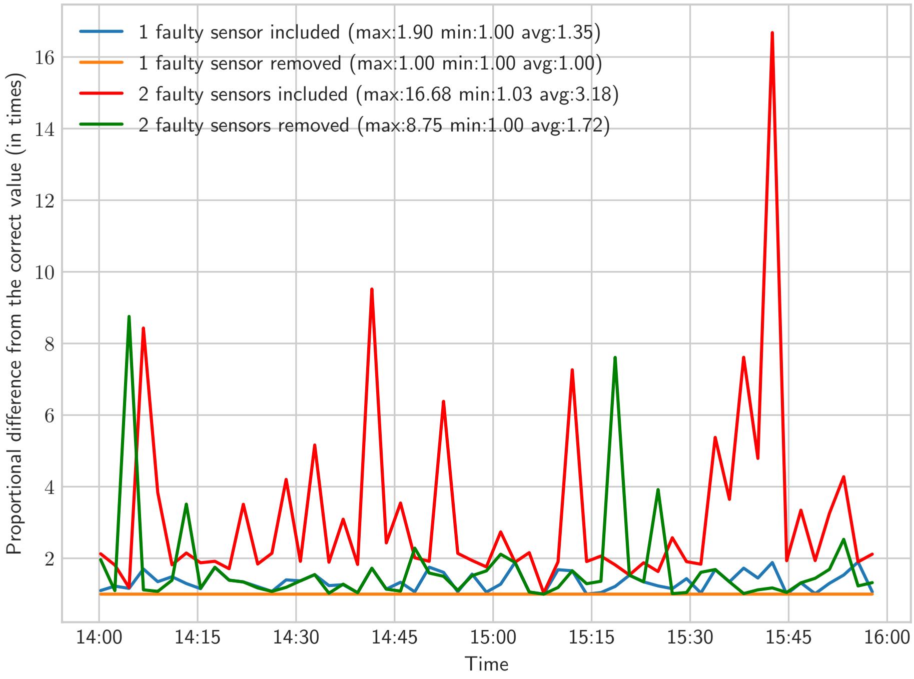


Figure 11: Proportional difference in aggregated data

In the following, the author shows real data (without any overriding) from two different naturally occurred incidents, where fault detection was performed from data at different layers.

5.3.2 Incident 1: Fault detection at the Raw Layer

Incident 1 illustrates a naturally occurred incident, which resulted in cables connecting the Sensor Nodes to the Sensor Gateway torn. Figure 12 shows data from one Sensor Group in different data layers during a 6 hour interval. The Aggregated data layer shows median water flow velocity over all the nodes from a Sensor Group at 5 meter depth in the given time period. The Processed data layer in Figure 12 shows Dominant Frequency on Sensors belonging to the same Sensor Group. The Raw Layer in Figure 12 shows the packet count of valid measurements during 180s intervals during this period. On the Raw data layer only one color can be seen, as all the drawn nodes follow the same line. Although the Sensor Groups consist of 4 Sensor Nodes, Node 004 was broken before the time of this event and was removed from the Sensor Group. Therefore, its measurements are not drawn. In Figure 12 all the layers clearly present an incident between 19:54-19:57, where correspondingly faults occurred on all data layers. The incident stopped incoming data from the Raw Layer, which is represented as the number of incoming raw data packets dropping to zero. The processed and aggregated layers show the last good values that remain constant because no new data came in after the incident. During the incident, the connection cables were physically cut (see Figure 13).

During the 6 hour period that is given around Incident 1, the events from the node level Health map are also quantified and shown in Table 4. The few fault events recorded were from the incident in which packets were disrupted, ending up with bad packet length and

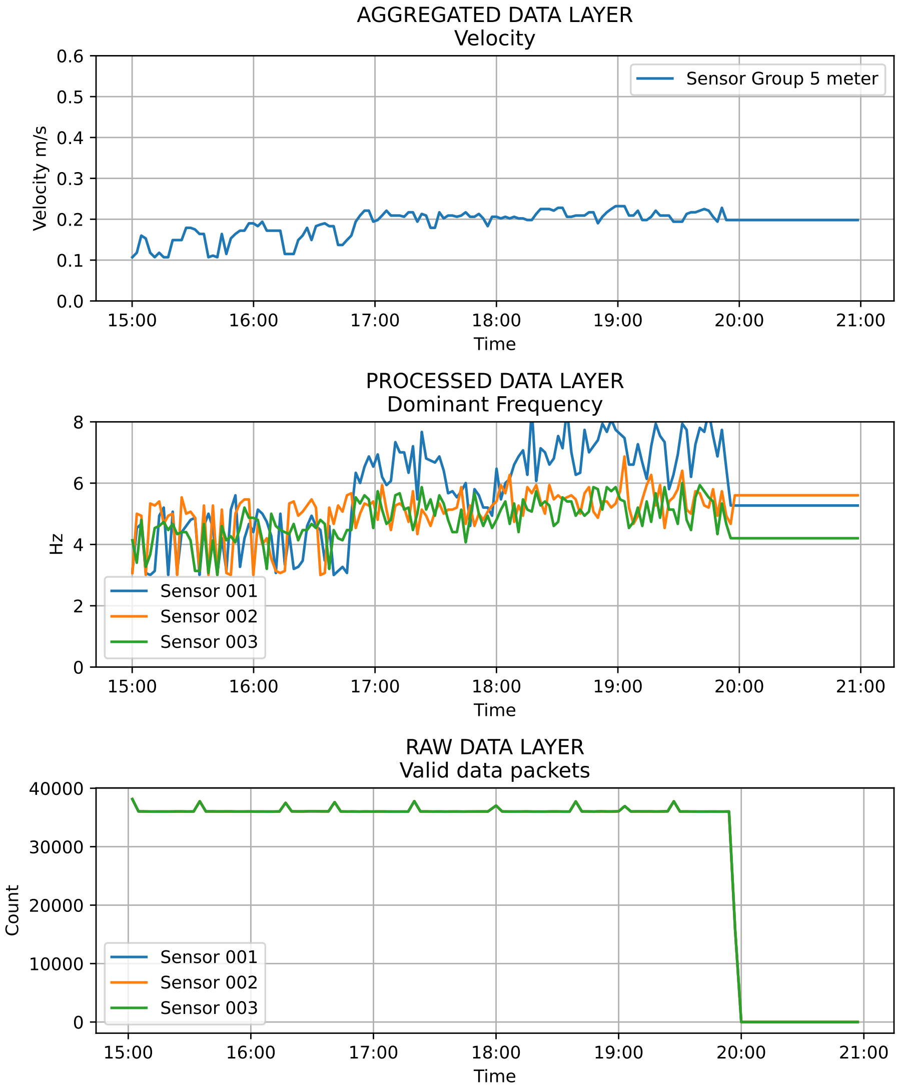


Figure 12: Incident 1 time series: Fault manifestation at the Data Layers

decoding errors. However, this error quantity is low and can be a part of normal operation. It can be seen that for this type of incident, fault detection can be performed from the Raw data layer monitoring the valid packet flow as shown in the lower section of Figure 12.

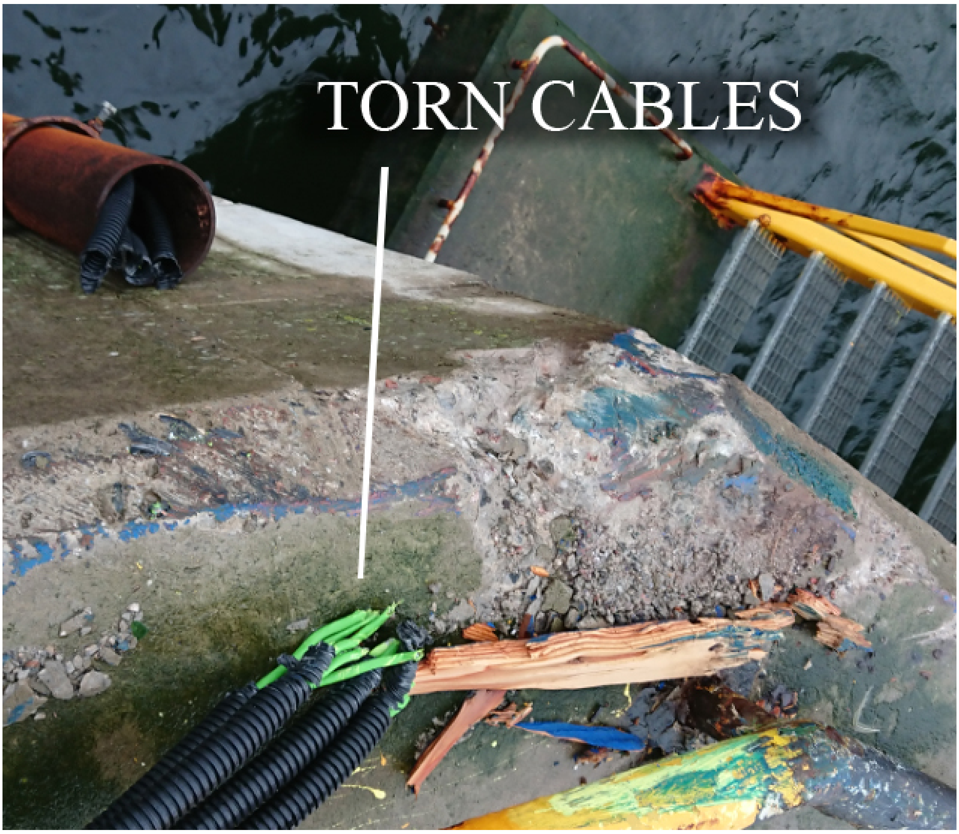


Figure 13: Incident 1: Incident damage

5.3.3 Incident 2: Fault Detection at the Processed Layer

Incident 2 represents an incident where a stem of a Sensor Node was detached. Figure 14 shows data from different data layers of one node during a 1-hour interval. In this case it can be seen that valid packets kept coming in, however the measurements dropped to zero. Thus $f_2(i,t)$ (fault on the processed layer L_2 of sensor i at time t) occurred, but no $f_1(i,t)$ (fault on the Raw data layer L_1 of sensor i at time t). It was later found that the moving part of a sensor's stem was torn off as shown in Figure 15. **It can be seen from Figure 14 that this type of incident was not identifiable from the Raw data layer, but rather from the sudden drop of the dominant frequency to zero at the Processed layer.**

Table 5 shows that there were no detected faults at the Raw layer ($\neg f_1(i,t)$) and the valid data packets count was higher than in Incident 1 (see Table 4) due to the fact that the data flow was never interrupted.

Table 4: *Incident 1 event quantification (same interval as Figure 12)*

Event type	**Count** **Sensor 001**	**Sensor 002**	**Sensor 003**
Valid packets	3559807	3559774	3559793
Data not yet ready	0	0	0
Bad packet length	3	2	2
Decoding error	0	1	0
Serial number changed	0	0	0
Port closed	0	0	0

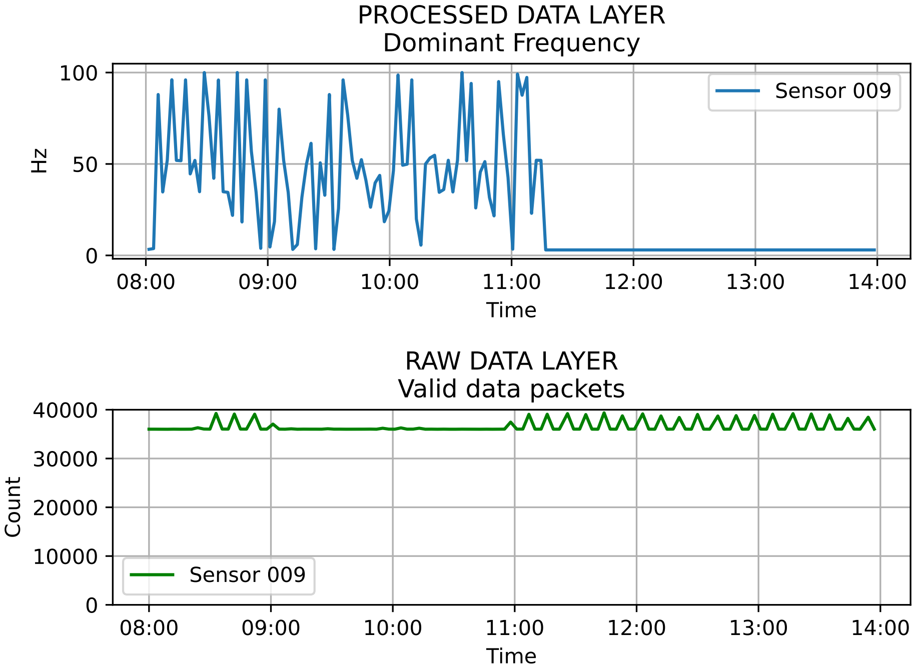


Figure 14: Incident 2 time series: Fault manifestation at the Data Layers

Table 5: *Incident 2 event quantification (same interval as Figure 14)*

Event type	**Count** **Sensor 009**
Valid packets	4317807
Data not yet ready	0
Bad packet length	0
Decoding error	0
Serial number changed	0
Port closed	0

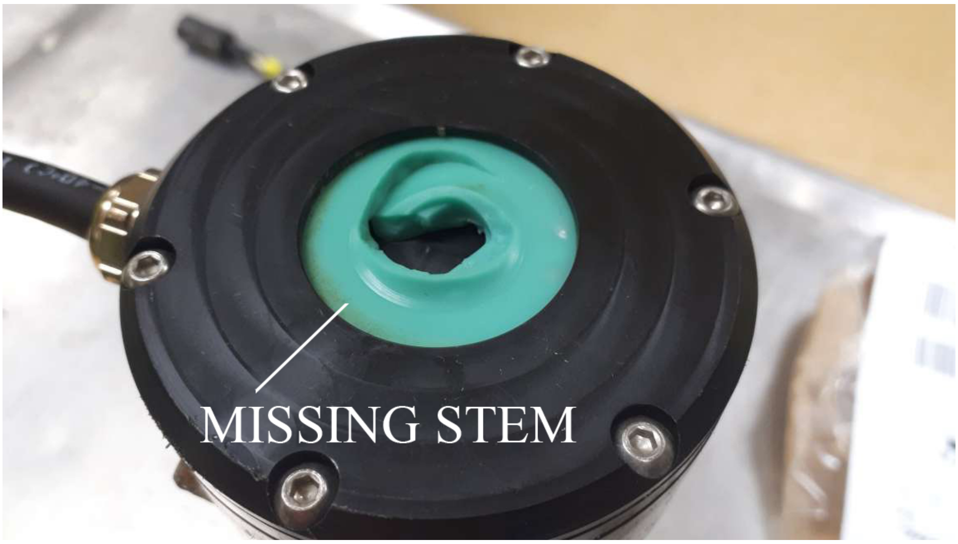


Figure 15: Incident 2: Incident damage

5.4 Chapter Summary

The proposed architecture was implemented and tested in real life on a single sensor network. Although our intention was to develop a generic sensor network architecture and it showed benefits in implemented water velocity measuring sensor network, it has not been validated on other type of real sensor networks.

The sensor network used for testing has essential signal processing needed for its functionality, so the data layers were not artificial, and data-driven cross-layer fault management was integral addition to the core functionality of the sensor network. Not in case of all types of sensor networks the proposed architecture may be possible to implement or be resource and cost efficient.

The incidents and synthetic scenarios based on real data show that the data-driven cross-layer fault management allows improving the accuracy of sensor group measurement by 35% in the case of single sensor errors and nearly twofold in case of double sensor errors. Additionally, the proposed architecture is cost and resource effective as it relies on the sensor network internal functionality and no additional hardware is needed. The architecture is scalable in the sense that there can be multiple hierarchies of sensor groups. In the future, it would be useful to implement and validate the same framework on different types of sensor network applications.

In the next chapter, fault management on the Aggregated data layer is researched to dynamically manage sensor outliers and neighboring sensor group degradation.

6 Fault Management in Data Aggregation

6.1 Motivation

The faults can also be managed on the Aggregated Data Layer. In the current chapter the author proposes a data-driven method for aggregating the sensor data to improve it's quality. The current chapter is based on the publication IV. One of the most popular mathematical state estimation tools is the Kalman Filter (KF) [46, 98, 45, 48, 49, 51, 69, 139, 82] with its multiple flavors. Sensor data aggregation can also be seen as a state estimation task, and KF is used for that [101],[9], [122]. The author uses KF data aggregation for fault masking with details described in the publication IV.

6.2 Methods

Due to physical limitations, harsh underwater environment and possibility of occurrence of both, persistent and intermittent faults, the sensors vary from correct measurements. To cope with that issue, the author relies on Sensor Groups (see Section 3.4) to generate data fusion for univariate measurements - that is multiple sensors simultaneously measure similar physical entity. The author is applying KF in the Aggregation Data layer (see section 3.4) after an initial signal processing is done.

The author is using adaptive KF for data fusion to compute aggregated data and get the filtered estimate that is more reliable than the sources. The KF sensor uncertainty matrices (see Publication IV for details) are updated in different configurations as explained below. The following KF configurations were implemented and compared:

- *Kalman Static*, where sensors' uncertainty is not updated.
- *Kalman Difference*, where sensors' uncertainty is updated using the difference function (see Section 6.2.1).
- *Kalman Latency*, where sensors' uncertainty is updated using latency function (See Section 6.2.1)
- *Kalman Adaptive*, where sensors' uncertainty is updated using both difference and latency functions.

The computational complexity of the updating steps of the KF state is dependent on the dimensionality of the measurement vector [123], which in our case is the number of sensors in a sensor group. Kalman Difference, Kalman Latency and Kalman Adaptive methods have additional computational steps for updating the uncertainty using residual and time values accordingly, making them more complex than the Kalman Static method.

The calculation is iterative over time and the aggregated values are the outputs of the KF. The aggregation processes of a sensor group are shown in Figure 16. At the predefined intervals the last known good processed values from all the sensors belonging to a single sensor group are collected, the sensor group's aggregated value is computed, and the next execution time scheduled.

6.2.1 Uncertainty from the Difference of Measurement and Estimation

Small latency for sensor measurements can be tolerated, but larger latency values deteriorate fast. For uncertainty caused by measurement latency, the author applied the sigmoid function described in the publication IV. Sigmoid function used for latency is shown in normalized form in Figure 17 (left).

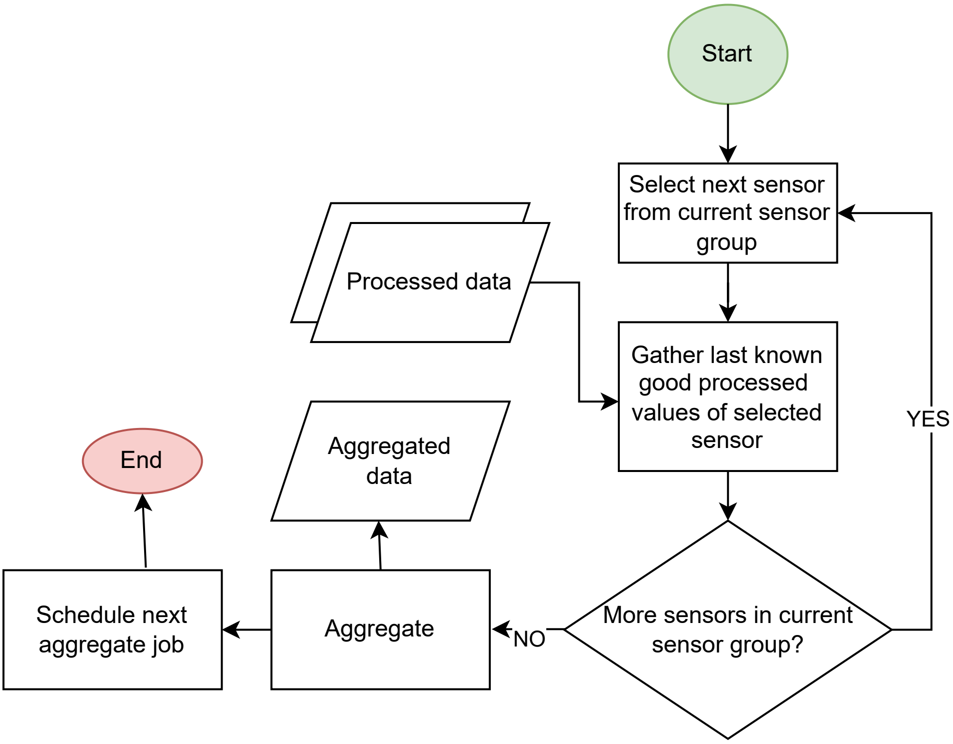

Start
Select next sensor from current sensor group
Processed data
Gather last known good processed values of selected sensor
YES
End
Aggregated data
More sensors in current sensor group?
NO
Schedule next aggregate job
Aggregate

Figure 16: Flowchart of the aggregation processes of a sensor group

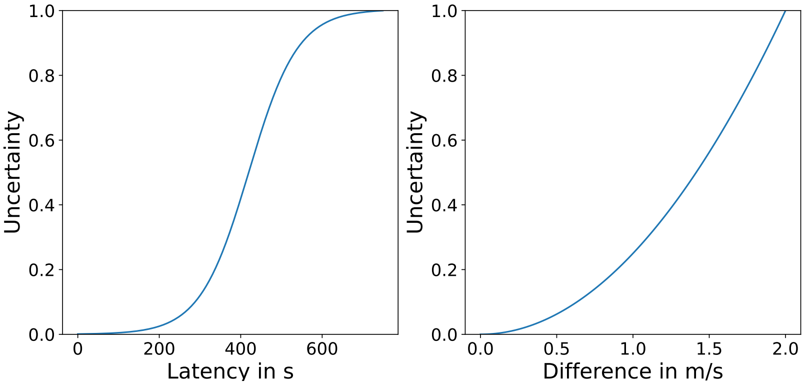


Figure 17: Left: (sigmoid function) Normalized sensor uncertainty from latency of the latest measurement; Right: (parabolic function) Normalized sensor uncertainty from measurement and prediction difference (residual)

Detecting outlier values based on the difference between estimation and measurement should not be linear - small differences in velocity should be proportionally more tolerated than larger differences. It should be noted that the sensors used had a defined reporting range, so the difference could not increase uncontrollably. For calculating adaptive weights based on difference, the author applied the parabole function that is described in publication IV. The parabole function used for the difference is shown in normalized form in Figure 17 (right).

6.3 Results

Experiments in two different underwater environments were conducted. The Flow Obstruction experiment was a short time experiment that took place in freshwater in a river on February 2, 2023. For this experiment, in addition to the Sensor Nodes, ADV (Acoustic Doppler Velocimeter) measurements were also used as reference values. The sensor network was installed to a river bed (see Section 6.3.1) and the water flow was manually disturbed and interfered with.

The Harbor Experiment was a long-time experiment active from April to August 2020. The sensor network was installed into sea water by a harbor for measuring underwater currents (see Section 3.2.1). For this experiment, we did not have a reference device. The sensors were neither disturbed nor interfered with manually, the collected data was naturally occurring. Most of the time during that period, the water flow was too slow to be measurable with sensors due to non-windy weather conditions. However, there were a couple of time intervals with a stronger water movement.

6.3.1 Flow Obstruction Experiment

The Sensor Nodes were attached to a metal bar at 20 cm intervals. Perpendicular to the center of the Sensor Node was another bar with attached ADV (Acoustic Doppler Velocimeter, Nortek Vectrino Profiler) approximately 50 cm from the Sensor Node metal bar. The construction was installed to a river bottom around 1m depth, with the ADV facing the flow and the Sensor Nodes side by side behind it. The order of the Sensor Nodes from the shore was H24, H25, H26, H27. The unobstructed water velocity appeared similar at all

Sensor Nodes. The Sensor Node offset coefficients are calibrated after installation to correspond to the mean value magnitude of the ADV beams. The velocity is calculated using the magnitude of median x and y axis angles of the 1 s time frame of 50 Hz measurements.

Figure 18 shows the obstruction experiment.

- Sensor measurements are shown as dots. Aggregations by different Kalman filters are shown as lines.
- A human was obstructing the flow by standing in the water for every Sensor Node for 30 s in the following order - 1st H24, 2nd H25, 3rd H26, 4th H27
- It can be seen that the obstruction changed the Sensor Node angles correspondingly as the dots representing single measurements move downwards at specific times.
- H25 obstruction is less clear, but happened also while obstructing H24 and H26, thus standing near the 1st and 3rd Sensor Node obstructed the flow also at the 2nd Sensor Node.

It can also be seen from Figure 18 that the water flow measurement of the sensors is consistent and reacts adequately to changes in the water flow. From Figure 18, it can be seen that the Kalman difference is the most optimal aggregation method for this case, as it accurately filters out disturbances at individual sensors. It was followed by median and Kalman adaptive aggregation. However, Kalman latency and Kalman static were far more influenced by disturbances at individual sensors.

6.3.2 Harbor Experiment

Finally an experiment was carried out on naturally occurring data from the actual use case. From the data collected during the five-month period, we selected an interval on 7 May 2020 where all of the sensors in the sensor group were active and there was enough flow to measure velocity and direction. The flow is in a measurable range from approximately 6:30 to 20:00 when it begins to fade. The day is characteristic for representing the harshness of the environment as there are multiple gaps and outliers in the readings, and one of the sensors (H3) stops providing new measurements around 13:20.

In this harsh, real-world environment, the Kalman difference and Kalman adaptive perform equally efficiently, while the aggregation provided by the Median and Kalman latency methods is far too unstable. The weakest performance is obtained by Kalman static, which is consistently overestimating the water current flow.

As the result of the experiments, the most robust and stable aggregation performance was achieved by the Kalman difference method, which took into account the difference of the measurement value at the sensor from the estimated value. This method performed well in a stable current as well as in the case of disturbances and also in very harsh conditions, where there were gaps and outliers.

Kalman adaptive was slightly less accurate with faulty sensor reading, but became more robust and equal to Kalman difference in case of more frequent gaps in readings. It might become the preferred option when conditions are extremely harsh and become even more dominated by gaps.

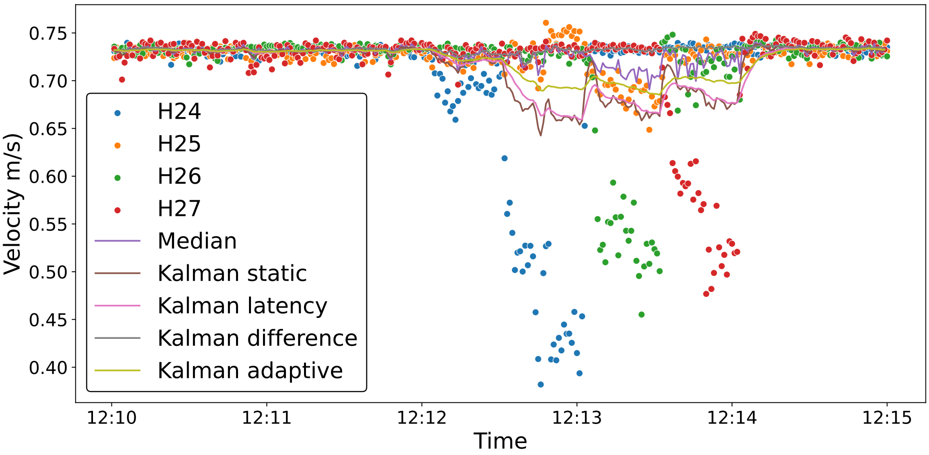


Figure 18: Comparison of the aggregation methods in the obstruction experiment

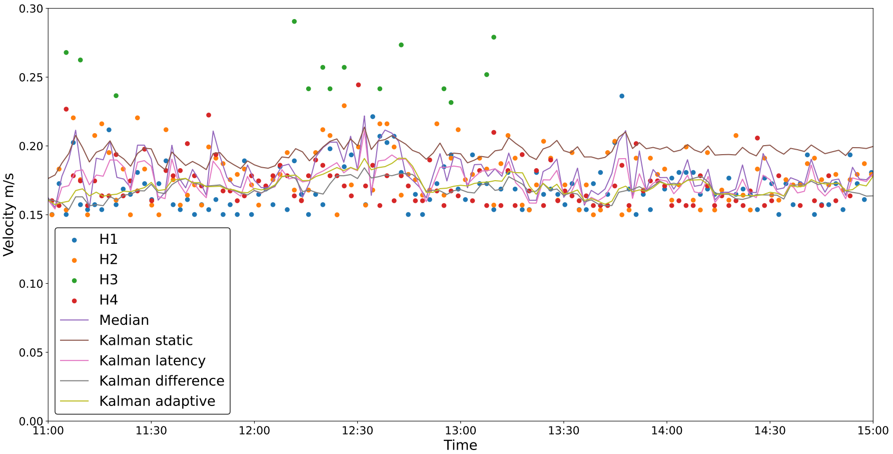


Figure 19: Aggregation methods and processed values for velocity measurement in the harbor experiment

6.4 Chapter Summary

A fault-resilient underwater sensor network based on sensor data aggregation by updating the measurement error matrix of an adaptive Kalman filter was proposed. A case study on a real-world harbor water flow monitoring use-case showed that the adaptive and difference based technique allowed for a significantly smoother aggregation in case of high fault rates in sensors' readings when compared to traditional Kalman filter and median value based aggregation techniques.

7 Conclusions

7.1 Summary

The current thesis proposed a data-driven and cross-layer resilient architecture for sensor networks, where instead of the system stack layers, the data layers are applied in fault detection and diagnosis to combine fault detection across data layers into a coordinated system health management architecture. The goal was to achieve graceful degradation in sensor networks, allowing them to continue their operations uninterrupted, but with a degraded quality of service. The main specific contributions of the thesis are summarized below.

- The thesis presented a systematic survey of fault-tolerant techniques in underwater sensor networks (USNs), collecting and categorizing the papers based on an introduced taxonomy of Fault Tolerance tasks and highlighting open research issues (addresses research question RQ1).
    - Systematic search of top papers and related papers was conducted and those papers analyzed.
    - Taxonomies for fault sources and Fault Tolerance tasks were described to categorize and systematize the analyzed papers.
    - The 127 analyzed papers were divided into categories, and open research issues as well as more addressed areas were identified.
    - The full table of the survey results was presented.
- The thesis also proposed a data-driven method to process sensor data improving the quality of data (RQ2).
    - An algorithm for in-door CO_2 base level correction in near real-time was developed based on CO_2 data logged during 6 months from 56 rooms.
    - An optimal trade-off between accuracy and data correction delay was reached using 1% percentile values in a 10 h sliding time window of the in-door CO_2 data.
    - The period with more than 1000ppm CO_2 concentration in classrooms after applying the developed baseline correction algorithm was almost 3 times less.
- The thesis proposed three data layers - Raw, Processed, Aggregated to use in data-driven cross-layer fault management (RQ3).
    - The sensor network architecture was proposed that is cost and resource effective as well as scalable, because it relies on the sensor network internal functionality with no additional hardware needed and hierarchies of sensor groups can be used.
    - Both synthetic scenarios and in-field, real-world experiments were provided for evaluating the cross-layer fault management capabilities and the aggregation of a system health map based on the faults manifesting at the different layers.
    - Data-driven cross-layer fault management allows the sensor group measurement accuracy to be improved by 35% in case of single sensor errors and nearly twice in case of double sensor error.

- A data-driven method was proposed to improve the quality of aggregated sensor data (RQ4).
  - An adaptive Kalman filter based data aggregation algorithm by updating the measurement error matrix was proposed. It was the first implementation where sigmoid and parabole functions were used for the adaptive Kalman filter weights, as far a the author knows,
  - A case study based on a real-world harbor water flow monitoring use-case was presented that showed the proposed method to allow for a significantly smoother aggregation in case of high fault rates in sensors' readings when compared to traditional Kalman filter and median value based aggregation techniques.
- A sensor network software framework was developed and implemented and experiments in natural environments were conducted.

7.2 Limitations and Future work

One of the limitations of the proposed architecture is that it may not be abstract enough, as it was validated in limited environments and mainly with one type of sensors - Hydromasts. In addition, data health history collection and prediction of the future state of an USN, for example, maintenance schedule planning, is not implemented nor discussed thoroughly.

To reduce the limitations in the future, the following directions can be further progressed:

- Different types of sensor network applications could be implemented. For used sensor networks the Raw, Processed and Aggregated data layers came naturally, while other sensor networks applications may have different requirements and the architecture framework may need to be shifted to fulfill those. This could help to make the framework more abstract and less dependent on the specific sensor network type.
- The architecture should be further tested based on data from different sources and environments. The sensors in a sensor group connected to a Sensor Hub in Edge environment and a Cloud service provider was considered generic enough to be enable to implement for various sources and environments, but it should be validated more with different sources.
- The architecture can be developed further to dynamically manage sensor outliers and degradation of sensor groups. Keeping track and storing the health of the single sensors and the degradation rate of the sensor groups may allow to implement capacity management and predicting the needed maintenance for the sensor groups.

List of Figures

1 Fault management in data layers presented in current thesis. 12
2 A radar chart of the analyzed papers addressing the main specific areas. ... 16
3 Taxonomy of Fault Tolerance tasks in USNs. 17
4 A) Sensor Node B) Harbor Installation 26
5 School building installation 27
6 Server-based Underwater Sensor Network framework software architecture. 28
7 The implemented underwater framework architecture. 31
8 Example of CO_2 concentration of all rooms during 5 weeks with initially logged (above) and corrected (below) baseline values 36
9 Cumulative CO_2 of all rooms in the same timeframe as previous figure 36
10 Raw and Processed data flowchart 39
11 Proportional difference in aggregated data 41
12 Incident 1 time series: Fault manifestation at the Data Layers 42
13 Incident 1: Incident damage 43
14 Incident 2 time series: Fault manifestation at the Data Layers 44
15 Incident 2: Incident damage 45
16 Flowchart of the aggregation processes of a sensor group 47
17 Left: (sigmoid function) Normalized sensor uncertainty from latency of the latest measurement; Right: (parabolic function) Normalized sensor uncertainty from measurement and prediction difference (residual) 48
18 Comparison of the aggregation methods in the obstruction experiment 50
19 Aggregation methods and processed values for velocity measurement in the harbor experiment 50

List of Tables

1 Categorized papers. 19
2 Data layers and their mapping to computing environments 32
3 The impact of baseline level correction of comparing logged values against different CO_2 concentration levels 37
4 Incident 1 event quantification (same interval as Figure 12) 44
5 Incident 2 event quantification (same interval as Figure 14) 44

References

[1] G. H. Adday, S. K. Subramaniam, Z. A. Zukarnain, and N. Samian. Fault Tolerance Structures in Wireless Sensor Networks (WSNs): Survey, Classification, and Future Directions. *Sensors*, 22(16):6041, aug 2022.

[2] K. A. Alansary, R. M. Daoud, H. H. Amer, and H. M. Elsayed. Networked Control System Architecture for Autonomous Underwater Vehicles with Redundant Sensors. *Proc. 11th Int. Conf. Electron. Comput. Artif. Intell. ECAI 2019*, pages 2019–2022, 2019.

[3] A. A. Amin and K. M. Hasan. A review of Fault Tolerant Control Systems: Advancements and applications. *Meas. J. Int. Meas. Confed.*, 143:58–68, 2019.

[4] A. Amory, B. Meyer, C. Osterloh, T. Tosik, and E. Maehle. Towards fault-tolerant and energy-efficient swarms of underwater robots. *Proc. - IEEE 27th Int. Parallel Distrib. Process. Symp. Work. PhD Forum, IPDPSW 2013*, pages 1550–1553, 2013.

[5] G. Ateniese, A. Capossele, P. Gjanci, C. Petrioli, and D. Spaccini. SecFUN: Security framework for underwater acoustic sensor networks. *MTS/IEEE Ocean. 2015 - Genova Discov. Sustain. Ocean Energy a New World*, 2015.

[6] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A survey. *Comput. Networks*, 54(15):2787–2805, oct 2010.

[7] S. P. Azad. *Cross-Layer Dependability Management in Network on Chip based System on Chip*. PhD thesis, 2018.

[8] J. Bar-Magen Numhauser. Fog computing introduction to a new cloud evolution. *Univ. Alcalá*, 2012.

[9] M. H. Basiri, N. L. Azad, and S. Fischmeister. Distributed time-varying kalman filter design and estimation over wireless sensor networks using OWA sensor fusion technique. *2020 28th Mediterr. Conf. Control Autom. MED 2020*, pages 325–330, 2020.

[10] L. Bauer, J. Henkel, A. Herkersdorf, M. A. Kochte, J. M. Kühn, W. Rosenstiel, T. Schweizer, S. Wallentowitz, V. Wenzel, T. Wild, H.-J. Wunderlich, and H. Zhang. Adaptive multi-layer techniques for increased system dependability. *it - Inf. Technol.*, 57(3):149–158, 2015.

[11] B. A. Begum and S. V. Nandury. Data aggregation protocols for WSN and IoT applications – A comprehensive survey. *J. King Saud Univ. - Comput. Inf. Sci.*, 35(2):651–681, 2023.

[12] G. Benincasa, G. D'Aniello, C. De Maio, V. Loia, and F. Orciuoli. Towards perception-oriented situation awareness systems. *Adv. Intell. Syst. Comput.*, 322:813–824, 2014.

[13] K. Benson. Enabling resilience in the Internet of Things. *2015 IEEE Int. Conf. Pervasive Comput. Commun. Work. PerCom Work. 2015*, pages 230–232, 2015.

[14] V. P. Bhuvana, C. Preissl, A. M. Tonello, and M. Huemer. Multi-Sensor Information Filtering with Information-Based Sensor Selection and Outlier Rejection. *IEEE Sens. J.*, 18(6):2442–2452, mar 2018.

[15] T. Bokareva, N. Bulusu, and S. Jha. SASHA: Toward a self-healing hybrid sensor network architecture. *Second IEEE Work. Embed. Networked Sensors, EmNetS-II*, 2005:71–78, 2005.

[16] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Pers. Commun.*, 7(5):28–34, 2000.

[17] L. Calderoni, A. Magnani, and D. Maio. IoT Manager: An open-source IoT framework for smart cities. *J. Syst. Archit.*, 98(December 2018):413–423, sep 2019.

[18] F. Campagnaro, R. Francescon, F. Guerra, F. Favaro, P. Casari, R. Diamant, and M. Zorzi. The DESERT underwater framework v2: Improved capabilities and extension tools. In *2016 IEEE Third Underw. Commun. Netw. Conf.*, pages 1–5. IEEE, aug 2016.

[19] M. Caporuscio, F. Flammini, N. Khakpour, P. Singh, and J. Thornadtsson. Smart-troubleshooting connected devices: Concept, challenges and opportunities. *Futur. Gener. Comput. Syst.*, 111:681–697, 2020.

[20] G. Cario, A. Casavola, P. Gjanci, M. Lupia, C. Petrioli, and D. Spaccini. Long lasting underwater wireless sensors network for water quality monitoring in fish farms. *Ocean. 2017 - Aberdeen*, 2017-Octob:1–6, 2017.

[21] N. P. Carter, H. Naeimi, and D. S. Gardner. Design techniques for cross-layer resilience. *2010 Des. Autom. Test Eur. Conf. Exhib. (DATE 2010)*, (February):1023–1028, 2010.

[22] Chae-Won Yun, Jae-Hoon Lee, Okyeon Yi, and Soo-Hyun Park. Ticket-based authentication protocol for Underwater Wireless Sensor Network. *2016 Eighth Int. Conf. Ubiquitous Futur. Networks*, pages 215–217, 2016.

[23] E. Cheng, S. Mirkhani, L. G. Szafaryn, C. Y. Cher, H. Cho, K. Skadron, M. R. Stan, K. Lilja, J. A. Abraham, P. Bose, and S. Mitra. CLEAR: Crosslayer exploration for architecting resilience combining hardware and software techniques to tolerate soft errors in processor cores. *Proc. - Des. Autom. Conf.*, 05-09-June:0–5, 2016.

[24] Z. Cheng, M. Perillo, and W. B. Heinzelman. General network lifetime and cost models for evaluating sensor network deployment strategies. *IEEE Trans. Mob. Comput.*, 7(4):484–497, 2008.

[25] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor. The SSN ontology of the W3C semantic sensor network incubator group. *J. Web Semant.*, 17:25–32, 2012.

[26] V. Cristea, C. Dobre, F. Pop, C. Stratan, A. Costan, C. Leordeanu, and E. Tirsa. A dependability layer for large-scale distributed systems. *Int. J. Grid Util. Comput.*, 2(2):109, 2011.

[27] X. Dai and Z. Gao. From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis. *IEEE Trans. Ind. Informatics*, 9(4):2226–2238, 2013.

[28] A. Dala, A. Adetomi, G. Enemali, and T. Arslan. RR4DSN: Reconfigurable Receiver for Deepwater Sensor Nodes. *2018 NASA/ESA Conf. Adapt. Hardw. Syst. AHS 2018*, pages 280–284, 2018.

[29] G. D'Aniello, A. Gaeta, and F. Orciuoli. Artificial bees for improving resilience in a sensor middleware for Situational Awareness. *TAAI 2015 - 2015 Conf. Technol. Appl. Artif. Intell.*, pages 300–307, 2016.

[30] E. Darra and S. K. Katsikas. A survey of intrusion detection systems in wireless sensor networks. *Intrusion Detect. Prev. Mob. Ecosyst.*, pages 393–458, 2017.

[31] A. P. Das and S. M. Thampi. Fault-resilient localization for underwater sensor networks. *Ad Hoc Networks*, 55:132–142, 2017.

[32] A. P. Das and S. M. Thampi. Fault-resilient localization for underwater sensor networks. *Ad Hoc Networks*, 55:132–142, 2017.

[33] R. de Lemos, C. Gacek, and A. Romanovsky. LNCS 3069 - Architecting Dependable Systems II. In *Archit. Dependable Syst. II*, pages 286–304. 2004.

[34] A. DeHon, H. M. Quinn, and N. P. Carter. Vision for cross-layer optimization to address the dual challenges of energy and reliability. *Proc. -Design, Autom. Test Eur. DATE*, (March):1017–1022, 2010.

[35] N. Desai and S. Punnekkat. Enhancing Fault Detection in Time Sensitive Networks using Machine Learning. *2020 Int. Conf. Commun. Syst. NETworkS, COMSNETS 2020*, pages 714–719, 2020.

[36] M. Díaz, C. Martín, and B. Rubio. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *J. Netw. Comput. Appl.*, 67:99–117, 2016.

[37] S. X. Ding. *Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems*. Advances in Industrial Control. Springer London, London, 2014.

[38] S. Distefano, G. Merlino, and A. Puliafito. A utility paradigm for IoT: The sensing Cloud. *Pervasive Mob. Comput.*, 20:127–144, jul 2015.

[39] M. C. Domingo. A topology reorganization scheme for reliable communication in underwater wireless sensor networks affected by shadow zones. *Sensors*, 9(11):8684–8708, 2009.

[40] M. C. Domingo. An overview of the internet of underwater things, nov 2012.

[41] M. C. Domingo and M. C. Vuran. Cross-layer analysis of error control in underwater wireless sensor networks. *Comput. Commun.*, 35(17):2162–2172, 2012.

[42] M. Dong, H. Li, Y. Li, Y. Deng, and R. Yin. Fault-tolerant topology with lifetime optimization for underwater wireless sensor networks. *Sadhana - Acad. Proc. Eng. Sci.*, 45(1), 2020.

[43] Y. Dong and H. Dong. Simulation study on cross-layer design for energy conservation in underwater acoustic networks. *Ocean. 2013 MTS/IEEE - San Diego An Ocean Common*, pages 0–4, 2013.

[44] Y. Dong, R. Wang, Z. Li, C. Cheng, and K. Zhang. Poster Abstract: Improved Reverse Localization Schemes for Underwater Wireless Sensor Networks. *2017 16th ACM/IEEE Int. Conf. Inf. Process. Sens. Networks*, 1(1):323–324, 2017.

[45] R. Doshmanziari, H. Khaloozadeh, and A. Nikoofard. Gas pipeline leakage detection based on sensor fusion under model-based fault detection framework. *J. Pet. Sci. Eng.*, 184(July 2019), 2020.

[46] J. Durbin and S. J. Koopman. *Time Series Analysis by State Space Methods*. Oxford University Press, may 2012.

[47] M. Egerer, A. Ristolainen, L. Piho, L. Vihman, and M. Kruusmaa. Hall effect sensor-based low-cost flow monitoring device: Design and validation. *IEEE Sensors Journal*, pages 1–1, 2024.

[48] P. J. Escamilla-Ambrosio and N. Mort. Multi-sensor data fusion architecture based on adaptive Kalman filters and fuzzy logic performance assessment. *Proc. 5th Int. Conf. Inf. Fusion, FUSION 2002*, 2(3):1542–1549, 2002.

[49] P. J. Escamilla-Ambrosio and N. Mort. Hybrid Kalman Filter-Fuzzy Logic Adaptive Multisensor Data Fusion Architectures. *Proc. IEEE Conf. Decis. Control*, 5(December):5215–5220, 2003.

[50] E. Fadel, V. C. Gungor, L. Nassef, N. Akkari, M. G. Abbas Malik, S. Almasri, and I. F. Akyildiz. A survey on wireless sensor networks for smart grid. *Comput. Commun.*, 71:22–33, 2015.

[51] E. Foxlin. Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter. *Proc. - Virtual Real. Annu. Int. Symp.*, pages 185–194, 1996.

[52] P. Freeman, R. Pandita, N. Srivastava, and G. J. Balas. Model-based and data-driven fault detection performance for a small UAV. *IEEE/ASME Trans. Mechatronics*, 18(4):1300–1309, 2013.

[53] G. Georgakos, U. Schlichtmann, R. Schneider, and S. Chakraborty. Reliability challenges for electric vehicles. In *Proc. 50th Annu. Des. Autom. Conf. - DAC '13*, page 1, New York, New York, USA, 2013. ACM Press.

[54] R. Golombek, S. Wrede, M. Hanheide, and M. Heckmann. Online data-driven fault detection for robotic systems. In *2011 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pages 3011–3016. IEEE, sep 2011.

[55] N. Goyal, M. Dave, and A. K. Verma. Data aggregation in underwater wireless sensor network: Recent approaches and issues. *J. King Saud Univ. - Comput. Inf. Sci.*, 2017.

[56] N. Goyal, M. Dave, and A. K. Verma. A novel fault detection and recovery technique for cluster-based underwater wireless sensor networks. *Int. J. Commun. Syst.*, 31(4):1–14, 2018.

[57] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of Things (IoT): A vision, architectural elements, and future directions. *Futur. Gener. Comput. Syst.*, 29(7):1645–1660, 2013.

[58] H. S. Gunawi, T. Do, J. M. Hellerstein, I. Stoica, D. Borthakur, and J. Robbins. Failure as a service (faas): A cloud service for large-scale, online failure drills. *Electr. Eng. Comput. Sci.*, pages 1–8, 2011.

[59] Y. Guo and Y. Liu. Localization for anchor-free underwater sensor networks. *Comput. Electr. Eng.*, 39(6):1812–1821, 2013.

[60] G. Han, Y. He, J. Jiang, H. Wang, Y. Peng, and K. Fan. Fault-Tolerant Trust Model for Hybrid Attack Mode in Underwater Acoustic Sensor Networks. *IEEE Netw.*, 34(5):330–336, 2020.

[61] G. Han, J. Jiang, N. Sun, and L. Shu. Secure communication for underwater acoustic sensor networks. *IEEE Commun. Mag.*, 53(8):54–60, aug 2015.

[62] G. Han, C. Zhang, L. Shu, and J. J. P. C. Rodrigues. Impacts of Deployment Strategies on Localization Performance in Underwater Acoustic Sensor Networks. *IEEE Trans. Ind. Electron.*, 62(3):1725–1733, mar 2015.

[63] Y. Han, J. Zhang, and D. Sun. Error control and adjustment method for underwater wireless sensor network localization. *Appl. Acoust.*, 130:293–299, 2018.

[64] J. Heidemann, Wei Ye, J. Wills, A. Syed, and Yuan Li. Research challenges and applications for underwater sensor networking. *IEEE Wirel. Commun. Netw. Conf. 2006. WCNC 2006.*, 00(c):228–235, 2006.

[65] A. A. Helal, R. S. Villaça, C. A. Santos, and R. Colistete. An integrated solution of software and hardware for environmental monitoring. *Internet of Things*, 19(November 2021):100518, aug 2022.

[66] J. Henkel, L. Bauer, H. Zhang, S. Rehman, and M. Shafique. Multi-Layer Dependability: From Microarchitecture to Application Level. *Proc. 51st Annu. Des. Autom. Conf. Des. Autom. Conf.*, pages 47:1–47:6, 2014.

[67] J. Henkel, L. Hedrich, A. Herkersdorf, R. Kapitza, D. Lohmann, P. Marwedel, M. Platzner, W. Rosenstiel, U. Schlichtmann, O. Spinczyk, M. Tahoori, L. Bauer, J. Teich, N. Wehn, H.-J. Wunderlich, J. Becker, O. Bringmann, U. Brinkschulte, S. Chakraborty, M. Engel, R. Ernst, and H. Härtig. Design and architectures for dependable embedded systems. In *Proc. seventh IEEE/ACM/IFIP Int. Conf. Hardware/software codesign Syst. Synth. - CODES+ISSS '11*, page 69, New York, New York, USA, 2011. ACM Press.

[68] C. J. Huang, Y. W. Wang, H. H. Liao, C. F. Lin, K. W. Hu, and T. Y. Chang. A power-efficient routing protocol for underwater wireless sensor networks. *Appl. Soft Comput. J.*, 11(2):2348–2355, 2011.

[69] T. Huang, H. Jiang, Z. Zou, L. Ye, and K. Song. An integrated adaptive Kalman filter for high-speed UAVs. *Appl. Sci.*, 9(9), 2019.

[70] R. J. Hyndman and Y. Fan. Sample Quantiles in Statistical Packages. *Am. Stat.*, 50(4):361, nov 1996.

[71] V. Isler, S. Kannan, and K. Daniilidis. Sampling Based Sensor-Network Deployment. pages 1780–1785, 2004.

[72] Information technology - Message Queuing Telemetry Transport (MQTT) v3.1.1. Standard, International Organization for Standardization, Geneva, CH, June 2016.

[73] M. Jahanbakht, W. Xiang, L. Hanzo, and M. R. Azghadi. Internet of Underwater Things and Big Marine Data Analytics - A Comprehensive Survey. *IEEE Commun. Surv. Tutorials*, 23(2):904–956, 2021.

[74] S. Jiang. State-of-the-Art Medium Access Control (MAC) Protocols for Underwater Acoustic Networks: A Survey Based on a MAC Reference Model. *IEEE Commun. Surv. Tutorials*, 20(1):96–131, 2018.

[75] Z. Jin, Q. Zhao, and Y. Luo. Routing Void Prediction and Repairing in AUV-Assisted Underwater Acoustic Sensor Networks. *IEEE Access*, 8:54200–54212, 2020.

[76] M. Kaaniche, J. C. Laprie, and J. P. Blanquart. Dependability engineering of complex computing systems. *Proc. IEEE Int. Conf. Eng. Complex Comput. Syst. ICECCS*, pages 36–46, 2000.

[77] C.-C. Kao, Y.-S. Lin, G.-D. Wu, and C.-J. Huang. A Comprehensive Study on the Internet of Underwater Things: Applications, Challenges, and Channel Models. *Sensors*, 17(7):1477, jun 2017.

[78] C.-C. Kao, Y.-S. Lin, G.-D. Wu, and C.-J. Huang. A study of applications, challenges, and channel models on the Internet of Underwater Things. In *2017 Int. Conf. Appl. Syst. Innov.*, number 2, pages 1375–1378. IEEE, may 2017.

[79] J. U. Khan and H. S. Cho. A multihop data-gathering scheme using multiple AUVs in hierarchical underwater sensor networks. *Int. Conf. Inf. Netw.*, 2016-March:265–267, 2016.

[80] M. Z. Khan. Fault Management in Wireless Sensor Networks. *GESJ Comput. Sci. Telecommun.*, 1(37), 2013.

[81] M. Z. Khan, M. Merabti, and B. Askwith. Design Considerations for Fault Management in Wireless Sensor Networks. (June), 2009.

[82] M. Kheirandish, E. A. Yazdi, H. Mohammadi, and M. Mohammadi. A fault-tolerant sensor fusion in mobile robots using multiple model Kalman filters. *Rob. Auton. Syst.*, 161, 2023.

[83] J. Y. Kim, H. Yoon, S. H. Kim, and S. H. Son. Fault management of robot software components based on OPRoS. *Proc. - 2011 14th IEEE Int. Symp. Object/Component/Service-Oriented Real-Time Distrib. Comput. ISORC 2011*, pages 253–260, 2011.

[84] Y. Koraz and H. a. Gabbar. Bayesian-neuro-fuzzy network based online condition monitoring system for resilient micro energy grid using FPGA. *2017 IEEE 7th Int. Conf. Power Energy Syst.*, pages 85–89, 2017.

[85] S. Kovács, P. Bühlmann, H. Li, and A. Munk. Seeded binary segmentation: a general methodology for fast and optimal changepoint detection. *Biometrika*, 110(1):249–256, feb 2023.

[86] M. Koval, M. Havlíček, and J. Tesař. General Sensors Network application approach. 12(1):1–5, 2023.

[87] P. Kuila and P. K. Jana. Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach. *Eng. Appl. Artif. Intell.*, 33:127–140, 2014.

[88] S. Kumar and Balamurugan B. Fault Tolerant Cloud Systems. In M. Khosrow-Pour, D.B.A., editor, *Encycl. Inf. Sci. Technol. Fourth Ed.*, chapter 93, pages 1075–1090. IGI Global, 4th edition, 2018.

[89] A. Lachenmann, P. Marron, D. Minder, M. Gauger, O. Saukh, and K. Rothermel. TinyXXL: Language and Runtime Support for Cross-Layer Interactions. In *2006 3rd Annu. IEEE Commun. Soc. Sens. Ad Hoc Commun. Networks*, number September, pages 178–187. IEEE, 2006.

[90] A. J. Lachenmann. *A Cross-Layer Framework for Sensor Networks*. PhD thesis, University of Stuttgart, 2008.

[91] C. Lal, R. Petroccia, M. Conti, and J. Alves. Secure underwater acoustic networks: Current and future research directions. *3rd Underw. Commun. Netw. Conf. Ucomms 2016*, 2016.

[92] M. H. Lee and Y. H. Choi. Fault detection of wireless sensor networks. *Comput. Commun.*, 31(14):3469–3475, 2008.

[93] J. Li, H. Gao, S. Zhang, S. Chang, J. Chen, and Z. Liu. Self-localization of autonomous underwater vehicles with accurate sound travel time solution. *Comput. Electr. Eng.*, 50:26–38, 2016.

[94] J. Liu, Z. Wang, J. H. Cui, S. Zhou, and B. Yang. A Joint Time Synchronization and Localization Design for Mobile Underwater Sensor Networks. *IEEE Trans. Mob. Comput.*, 15(3):530–543, 2016.

[95] T.-H. Liu, S.-C. Yi, and X.-W. Wang. A fault management protocol for low-energy and efficient Wireless sensor networks. *J. Inf. Hiding Multimed. Signal Process.*, 4(1):34–45, 2013.

[96] J. López-Riquelme, N. Pavón-Pulido, H. Navarro-Hellín, F. Soto-Valles, and R. Torres-Sánchez. A software architecture based on FIWARE cloud for Precision Agriculture. *Agric. Water Manag.*, 183:123–135, mar 2017.

[97] D. Lorenz, M. Barke, and U. Schlichtmann. Efficiently analyzing the impact of aging effects on large integrated circuits. *Microelectron. Reliab.*, 52(8):1546–1552, aug 2012.

[98] S. Mellah, G. Graton, E. M. E. Adell, M. Ouladsine, and A. Planchais. Mobile robot additive fault diagnosis and accommodation. *2019 8th Int. Conf. Syst. Control. ICSC 2019*, pages 241–246, 2019.

[99] M. Melo and G. Aquino. FaTEMa: A Framework for Multi-Layer Fault Tolerance in IoT Systems. *Sensors*, 21(21):7181, oct 2021.

[100] Y. Mengjie, H. Mokhtar, and M. Merabti. Fault management in wireless sensor networks. *IEEE Wirel. Commun.*, 14(6):13–19, 2007.

[101] K. Miao, W. A. Zhang, and X. Qiu. An adaptive unscented Kalman filter approach to secure state estimation for wireless sensor networks. *Asian J. Control*, 25(1):629–636, 2023.

[102] S. Mitra. Comparative Study of Fault Recovery Techniques in Wireless Sensor Network. (December):19–21, 2016.

[103] S. Mitra, K. Brelsford, and P. N. Sanda. Cross-layer resilience challenges: Metrics and optimization. *2010 Des. Autom. Test Eur. Conf. Exhib. (DATE 2010)*, (March 2010):1029–1034, 2010.

[104] N. Mohamed, I. Jawhar, J. Al-Jaroodi, and L. Zhang. Sensor network architectures for monitoring underwater pipelines. *Sensors*, 11(11):10738–10764, 2011.

[105] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *Int. J. Surg.*, 8(5):336–341, 2010.

[106] A. More and V. Raisinghani. A survey on energy efficient coverage protocols in wireless sensor networks. *J. King Saud Univ. - Comput. Inf. Sci.*, 29(4):428–448, 2017.

[107] E. Moridi, M. Haghparast, M. Hosseinzadeh, and S. J. Jassbi. Fault management frameworks in wireless sensor networks: A survey. *Comput. Commun.*, 155(February):205–226, 2020.

[108] E. Mortazavi, R. Javidan, M. J. Dehghani, and V. Kavoosi. A robust method for underwater wireless sensor joint localization and synchronization. *Ocean Eng.*, 137(September 2016):276–286, 2017.

[109] A. Mylonas, O. B. Kazanci, R. K. Andersen, and B. W. Olesen. Capabilities and limitations of wireless CO2, temperature and relative humidity sensors. *Build. Environ.*, 154:362–374, may 2019.

[110] M. R. Napolitano, C. Neppach, V. Casdorph, S. Naylor, M. Innocenti, and G. Silvestri. Neural-network-based scheme for sensor failure detection, identification, and accommodation. *J. Guid. Control. Dyn.*, 18(6):1280–1286, nov 1995.

[111] S. Nassif. Modeling and analysis of manufacturing variations. *Proc. IEEE 2001 Cust. Integr. Circuits Conf. (Cat. No.01CH37169)*, pages 223–228, 2001.

[112] C. Neti, M. H. Schneider, and E. D. Young. Maximally Fault Tolerant Neural Networks. *IEEE Trans. Neural Networks*, 3(1):14–23, 1992.

[113] Y. Noh, U. Lee, S. Lee, P. Wang, L. F. Vieira, J. H. Cui, M. Gerla, and K. Kim. HydroCast: Pressure routing for underwater sensor networks. *IEEE Trans. Veh. Technol.*, 65(1):333–347, 2016.

[114] S. M. Oteafy and H. S. Hassanein. Resilient Sensor Networks utilizing dynamic resource reuse: A novel paradigm towards fault-tolerant wireless sensing. *2014 Int. Conf. Comput. Manag. Telecommun. ComManTel 2014*, pages 217–222, 2014.

[115] B. Otto and M. Hompel. *Designing Data Spaces*. Springer International Publishing, Cham, 2022.

[116] L. Paradis and Q. Han. A survey of fault management in wireless sensor networks. *J. Netw. Syst. Manag.*, 15(2):171–190, 2007.

[117] S. Paul, J. Pan, and R. Jain. Architectures for the future networks and the next generation Internet: A survey. *Comput. Commun.*, 34(1):2–42, 2011.

[118] C. Petrioli, R. Petroccia, J. R. Potter, and D. Spaccini. The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks. *Ad Hoc Networks*, 34:224–238, 2015.

[119] R. Petroccia and D. Spaccini. Comparing the SUNSET and DESERT frameworks for in field experiments in underwater acoustic networks. *Ocean. 2013 MTS/IEEE Bergen Challenges North. Dimens.*, 2013.

[120] A. Prasanth. Certain Investigations on Energy-Efficient Fault Detection and Recovery Management in Underwater Wireless Sensor Networks. *J. Circuits, Syst. Comput.*, 2020.

[121] W. Qi, Y. Xia, S. Zhang, S. Zhang, and L. Zhu. Research on Stability-Enhanced Clustering Algorithm Based on Distributed Node Status Judgment in MWSN. *Electron.*, 11(23):1–19, 2022.

[122] J. Qin, J. Wang, L. Shi, and Y. Kang. Randomized Consensus-Based Distributed Kalman Filtering over Wireless Sensor Networks. *IEEE Trans. Automat. Contr.*, 66(8):3794–3801, 2021.

[123] M. Raitoharju and R. Piche. On Computational Complexity Reduction Methods for Kalman Filter Extensions. *IEEE Aerosp. Electron. Syst. Mag.*, 34(10):2–19, oct 2019.

[124] S. Rani, S. Hassan, J. Malhotra, and R. Talwar. Energy e ffi cient chain based routing protocol for underwater wireless sensor networks. *J. Netw. Comput. Appl.*, 92(December 2016):1–9, 2017.

[125] M. Rätsep, K. E. Parnell, T. Soomere, M. Kruusmaa, A. Ristolainen, and J. A. Tuhtan. Surface vessel localization from wake measurements using an array of pressure sensors in the littoral zone. *Ocean Eng.*, 233:109156, aug 2021.

[126] T. Rault, A. Bouabdallah, and Y. Challal. Energy efficiency in wireless sensor networks: A top-down survey. *Comput. Networks*, 67:104–122, 2014.

[127] S. Rehman. *Reliable Software for Unreliable Hardware – A Cross-Layer Approach*. Doctoral dissertation, Karlsruhe Institute of Technology (KIT), 2015.

[128] S. Rehman, K. H. Chen, F. Kriebel, A. Toma, M. Shafique, J. J. Chen, and J. Henkel. Cross-Layer Software Dependability on Unreliable Hardware. *IEEE Trans. Comput.*, 65(1):80–94, 2016.

[129] REHVA. REHVA COVID-19 guidance document v3, v4.1.

[130] A. Ristolainen, K. Kalev, J. A. Tuhtan, A. Kuusik, and M. Kruusmaa. Hydromorphological Classification Using Synchronous Pressure and Inertial Sensing. *IEEE Trans. Geosci. Remote Sens.*, 56(6):3222–3232, jun 2018.

[131] A. Ristolainen, L. Piho, and M. Kruusmaa. Feasibility study on distributed flow sensing with inertial sensors in aquaculture fish cages. *Aquac. Eng.*, 98(June):102271, aug 2022.

[132] A. Ristolainen, J. A. Tuhtan, and M. Kruusmaa. Continuous, near-bed current velocity estimation using pressure and inertial sensing. *IEEE Sens. J.*, PP(c):1–1, 2019.

[133] R. Roman, P. Najera, and J. Lopez. Securing the Internet of things. *Computer (Long. Beach. Calif).*, 44(9):51–58, sep 2011.

[134] P. S. Rossi, S. Member, and D. Ciuonzo. Energy Detection for MIMO Decision Fusion in Underwater Sensor Networks. *IEEE Sens. J.*, 15(3):1630–1640, 2015.

[135] S. S. Sahoo, B. Veeravalli, and A. Kumar. Cross-layer fault-tolerant design of real-time systems. In *2016 IEEE Int. Symp. Defect Fault Toler. VLSI Nanotechnol. Syst. DFT 2016*, pages 63–68. IEEE, sep 2016.

[136] B. Sahu, P. Sahu, and P. Dash. Fault Tolerant Dynamic Multi-Hop Clustering in Under Water Sensor Network. In *2018 Int. Conf. Inf. Technol.*, pages 101–105. IEEE, dec 2018.

[137] I. Salera, A. Agbaria, and M. Eltoweissy. Fault-tolerant mobile sink in networked sensor systems. *2006 2nd IEEE Work. Wirel. Mesh Networks, WiMESH 2006*, pages 106–108, 2007.

[138] P. Salhofer. Evaluating the FIWARE Platform. volume 9, 2018.

[139] J. Z. Sasiadek and P. Hartana. Sensor data fusion using Kalman filter. *Proc. 3rd Int. Conf. Inf. Fusion, FUSION 2000*, 2:19–25, 2000.

[140] Z. Sauli, V. Retnasamy, S. Taniselass, A. H. Shapri, R. M. Hatta, and M. H. Aziz. Polymer core BGA vertical stress loading analysis. *Proc. Int. Conf. Comput. Intell. Model. Simul.*, 129:148–151, 2012.

[141] M. L. Seto and A. Z. Bashir. Fault tolerance considerations for long endurance AUVs. *Proc. - Annu. Reliab. Maintainab. Symp.*, 2017.

[142] M. Shah, Z. Wadud, A. Sher, M. Ashraf, Z. A. Khan, and N. Javaid. Position adjustment-based location error-resilient geo-opportunistic routing for void hole avoidance in underwater sensor networks. *Concurr. Comput.*, (February):1–18, 2018.

[143] S. P. Singh and S. C. Sharma. A survey on cluster based routing protocols in wireless sensor networks. *Procedia Comput. Sci.*, 45(C):687–695, 2015.

[144] S. A. Sofi and R. N. Mir. Natural algorithm based adaptive architecture for underwater wireless sensor networks. *Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017*, 2018-Janua:2343–2346, 2018.

[145] M. Suvarna, A. Patil, and M. P. Mishra. Improved Mobicast Routing Protocol To Minimize Energy Consumption for Underwater Sensor Networks. (April), 2017.

[146] B. Tanasa, U. D. Bordoloi, P. Eles, and Z. Peng. Reliability-aware frame packing for the static segment of flexray. In *Proc. ninth ACM Int. Conf. Embed. Softw.*, pages 175–184, New York, New York, USA, 2011. ACM Press.

[147] A. S. Tanenbaum and M. Van Steen. *Distributed systems: principles and paradigms*. Prentice-Hall, 2007.

[148] Y. Tang, Y. Chen, W. Yu, and X. Xu. Emergency Communication Schemes for Mulitihop Underwater Acoustic Cooperartive Networks. *2018 IEEE Int. Conf. Signal Process. Commun. Comput. ICSPCC 2018*, (41476026), 2018.

[149] S. Temp. Indoor Air Quality Monitor SMT-IAQ3 manual.

[150] A. Y. Teymorian, S. Member, W. Cheng, L. Ma, and X. Cheng. 3D Underwater Sensor Network Localization. *Network*, 8(12):1610–1621, 2009.

[151] A. Thomas and K. Pattabiraman. Error detector placement for soft computation. *Proc. Int. Conf. Dependable Syst. Networks*, 15(1):1–25, jan 2013.

[152] S. Tyagi and N. Kumar. A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks. *J. Netw. Comput. Appl.*, 36(2):623–645, 2013.

[153] V. D. Valerio, C. Petrioli, L. Pescosolido, and M. Van Der Shaar. A Reinforcement Learning-based Data-Link Protocol for Underwater Acoustic Communications. *Proc. 10th Int. Conf. Underw. Networks Syst. - WUWNET '15*, pages 1–5, 2015.

[154] M. Veleski, R. Kraemer, and M. Krstic. An Overview of Cross-Layer Resilience Design Methods. In *Fifth Int. Conf. Radiat. Appl. Var. Fields Res.*, 2017.

[155] L. Vihman, M. Kruusmaa, and J. Raik. Data-Driven Cross-Layer Fault Management Architecture for Sensor Networks. In *2020 16th Eur. Dependable Comput. Conf.*, pages 33–40. IEEE, sep 2020.

[156] L. Vihman, M. Kruusmaa, and J. Raik. Systematic Review of Fault Tolerant Techniques in Underwater Sensor Networks. *Sensors*, 21(9):3264, may 2021.

[157] L. Vihman, T. M. Parts, H. K. Aljas, M. Thalfeldt, and J. Raik. Algorithms for online CO2 baseline correction in intermittently occupied rooms. In *18th Heal. Build. Eur. Conf.*, Aachen, Germany, 2023. Emerald.

[158] L. Vihman and J. Raik. Adaptive Kalman Filter Based Data Aggregation in Fault-Resilient Underwater Sensor Networks. In *2023 24th Int. Conf. Digit. Signal Process.*, pages 1–5. IEEE, jun 2023.

[159] K. Wang, H. Gao, X. Xu, J. Jiang, and D. Yue. An Energy-Efficient Reliable Data Transmission Scheme for Complex Environmental Monitoring in Underwater Acoustic Sensor Networks. *IEEE Sens. J.*, 16(11):4051–4062, 2016.

[160] Y. Wang, L. Cao, and T. A. Dahlberg. Efficient fault tolerant topology control for three-dimensional wireless networks. *Proc. - Int. Conf. Comput. Commun. Networks, ICCCN*, pages 336–341, 2008.

[161] C. H. Wu, K. C. Lee, and Y. C. Chung. A Delaunay Triangulation based method for wireless sensor network deployment. *Comput. Commun.*, 30(14-15):2744–2752, 2007.

[162] L. Xing. Reliability in Internet of Things: Current Status and Future Perspectives. *IEEE Internet Things J.*, 7(8):6704–6721, 2020.

[163] J. Xu, K. Li, and G. Min. Reliable and energy-efficient multipath communications in underwater sensor networks. *IEEE Trans. Parallel Distrib. Syst.*, 23(7):1326–1335, 2012.

[164] M. Xu and G. Liu. Fault tolerant routing in three-dimensional underwater acoustic sensor networks. *2011 Int. Conf. Wirel. Commun. Signal Process. WCSP 2011*, 2011.

[165] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter. Drone networks: Communications, coordination, and sensing. *Ad Hoc Networks*, 68:1–15, 2018.

[166] C. H. Yu, K. H. Lee, H. P. Moon, J. W. Choi, and Y. B. Seo. Sensor Localization Algorithms in Underwater Wireless Sensor Networks. pages 1760–1764, 2009.

[167] N. Z. Zenia, M. Aseeri, M. R. Ahmed, Z. I. Chowdhury, and M. Shamim Kaiser. Energy-efficiency and reliability in MAC and routing protocols for underwater wireless sensor network: A survey. *J. Netw. Comput. Appl.*, 71:72–85, 2016.

[168] Y. J. Zhao, R. Govindan, and D. Estrin. Residual energy scan for monitoring sensor networks. In *IEEE Wirel. Commun. Netw. Conf. WCNC*, volume 1, pages 356–362. IEEE, 2002.

[169] W. Zhehao and L. Xia. An Improved Underwater Acoustic Network Localization Algorithm. *China Commun.*, 12(3):77 – 83, 2015.

[170] A. Zhirabok and S. Pavlov. Data-driven method of fault detection in technical systems. *Procedia Eng.*, 100(January):242–248, 2015.

[171] Z. Zhou, B. Yao, R. Xing, L. Shu, and S. Bu. E-CARP: An Energy Efficient Routing Protocol for UWSNs in the Internet of Underwater Things. *IEEE Sens. J.*, 16(11):4072–4082, 2016.

[172] Y. Zhu, X. Lu, L. Pu, Y. Su, R. Martin, M. Zuba, Z. Peng, and J.-h. Cui. Aqua-Sim : An NS-2 Based Simulator for Underwater Sensor Networks. page 2014, 2014.

[173] Z. A. Zukarnain, O. A. Amodu, C. Wenting, and U. A. Bukar. A Survey of Sybil Attack Countermeasures in Underwater Sensor and Acoustic Networks. *IEEE Access*, 11(May):64518–64543, 2023.

Acknowledgements

I’d like to thank my supervisors for the help and patience, my family and the wonderful colleagues from the Centre for Biorobotics and the Centre of Dependable Computing.

Abstract

Data-Driven Fault-Resilient Cross-Layer Sensor Network Architecture

Underwater Sensor Networks (USNs) have found extensive application in environments fraught with challenges unique to underwater conditions, such as extreme pressures, limited visibility, and communication constraints. These networks play a pivotal role in applications ranging from harbor security to underwater pipeline monitoring. This thesis is motivated by the critical need to develop fault management techniques for USNs, where faults can lead to inaccurate data, system failures, and severe consequences, particularly in safety-critical scenarios.

The underwater environment presents numerous hazards, including manufacturing defects in hardware components, wear and tear due to continuous use and environmental factors, power supply issues, and the potential for human errors during installation and maintenance. Anticipating an increase in hardware faults with technological advances, this research explores cross-layer fault tolerance as a means of coping with imperfections in system components.

The primary research questions addressed by this thesis include leveraging sensor data for fault management and graceful degradation, implementing cross-layer fault management without significant hardware costs, applying real-time fault detection and localization techniques to sensor data, ensuring the reliability of sensor data integrity, adapting processing algorithms to changing conditions or faults, and validating the quality of aggregated data from multiple sensors.

The thesis introduces a data-driven approach that utilizes inherent redundancy in sensor networks, proposing data layers (raw, processed, and aggregated) for efficient cross-layer fault management. Contributions include a comprehensive survey of fault-tolerant techniques in USNs, data-driven fault management methods that allow to improve measurement accuracy by 35% in case of single sensor faults and nearly double in case of a double sensor faults, and a novel approach to sensor data aggregation. The proposed techniques are evaluated in the context of a challenging harbor USN with unreliable sensor readings.

This research advances fault management strategies crucial for maintaining the reliability, accuracy, and safety of sensor networks. By proactively managing faults, these networks can continue to operate efficiently and provide accurate data, making them invaluable for a wide range of applications.

Kokkuvõte
Andmepõhine tõrkekindel kihtideülene sensorvõrgu arhitektuur

Veealused sensorvõrgud on leidnud laialdast rakendust karmides keskkondades, mis on omased veealustele oludele ning täis väljakutseid nagu äärmuslik rõhk, piiratud nähtavus ja sidepiirangud. Need võrgustikud mängivad olulist rolli alates sadama turvalisusest kuni veealuste torujuhtmete seireni. Käesolev lõputöö on motiveeritud kriitilisest vajadusest töötada välja rikkehaldustehnikad veealuste sensorvõrkude jaoks, kus rikked võivad põhjustada ebatäpseid andmeid, süsteemi tõrkeid ja eriti tõsiseid tagajärgi ohutuskriitiliste stsenaariumide korral.

Veealuses keskkonnas asuvaid sensorvõrke mõjutab terve rida ohte, sealhulgas riistvarakomponentide tootmisdefektid, pidevast kasutamisest ja keskkonnast tingitud kulumine, toiteallika probleemid ja võimalikud inimlikud vead paigaldamise ja hoolduse ajal. Tehnoloogia arengust tulenevalt ennustatakse riistvararikete kasvu. Käesolev töö uurib kihtidevahelist rikketaluvust kui vahendit puudustega toimetulekuks süsteemi komponentides.

Väitekirjas käsitletavad peamised uurimisküsimused hõlmavad andurite andmete võimendamist rikete haldamiseks ja nendega toimetulemiseks, rakendades kihtidevahelist rikkehaldust ilma oluliste riistvarakuludeta, rakendades reaalajas rikete tuvastamist ja lokaliseerimist. anduri andmetele, tagades anduriandmete terviklikkuse usaldusväärsuse, kohandades töötlemisalgoritmid muutuvate tingimuste või tõrgete jaoks ja valideerides mitmetest anduritest agregeeritud andmeid.

Doktoritöö tutvustab andmepõhist lähenemist, mis kasutab sensorvõrkudele omast liiasust, pakkudes välja andmekihid (toorandmed, töödeldud ja agregeeritud andmed) tõhusaks kihtideüleseks rikkehalduseks. Töö teaduspanus hõlmab põhjalikke tõrketaluvustehnikaid allvee sensorvõrkudes, andmepõhiseid rikkehaldusmeetodeid, mis võimaldavad ühe anduri rikke korral suurendada mõõtetäpsust 35% ja kahe anduri rikke korral peaaegu kahekordistada, ning ka uudset lähenemist andurite andmete agregeerimiseks. Välja töötatud tehnikaid hinnati sadamas, reaalses keskkonnas, ebausaldusväärsete andurinäitudega.

Käesolev uuring edendab rikete haldamise strateegiaid, mis on andurite võrkude usaldusväärsuse, täpsuse ja ohutuse säilitamiseks üliolulised. Rikete ennetav haldus tagab, et need võrgud saavad jätkuvalt tõhusalt töötada ja pakkuda täpseid andmeid, muutes need sobivaks mitmete rakenduste jaoks.

Appendix 1

I

L. Vihman, M. Kruusmaa, and J. Raik. Data-Driven Cross-Layer Fault Management Architecture for Sensor Networks. In *2020 16th Eur. Dependable Comput. Conf.*, pages 33–40. IEEE, sep 2020

Data-Driven Cross-Layer Fault Management Architecture for Sensor Networks

Lauri Vihman
Department of Computer Systems
Tallinn University of Technology
Tallinn, Estonia
lauri.vihman@taltech.ee

Maarja Kruusmaa
Department of Computer Systems
Tallinn University of Technology
Tallinn, Estonia
maarja.kruusmaa@taltech.ee

Jaan Raik
Department of Computer Systems
Tallinn University of Technology
Tallinn, Estonia
jaan.raik@taltech.ee

***Abstract*—The paper proposes a data-driven cross-layer resilient architecture for sensor networks. The novelty of the approach lies in combining fault detection across data and network layers into a coordinated system health management architecture. The implemented fault detection is entirely data-driven: data are collected exclusively by the functional sensors that are part of the system. Thus, there is no need for additional hardware resources. The data layers considered include the raw sensor data layer, the processed data layer and the data aggregation layer. The proposed cross-layer fault management architecture utilizes a hierarchical health-map structure for fault detection and data aggregation. A practical case study of an underwater sensor network for harbor water flow monitoring application based on the proposed architecture is presented. Synthetic experiments with real data demonstrate the effectiveness of the approach in fault detection and diagnosis. The experiments show that the data-driven cross-layer fault management allows improving the sensor group measurement accuracy by 35% in case of single sensor errors and nearly twofold in case of double sensor errors. The paper also presents examples of system health-map aggregation and fault diagnosis based on faults manifesting at the different layers for real incidents occurring in the field.**

***Index Terms*—data-driven, cross-layer, fault management, fault resilience, sensor network, under-water**

I. INTRODUCTION

Sensor networks are a promising solution for monitoring the physical environment and providing observations for various applications in the cyber-physical critical infrastructure [1]. However, they often operate in harsh environments causing reliability issues in their sensing, computing and communication. Moreover, the growing complexity of sensor networks is expected to increase the frequency of faults [2]. At the same time, due to cost considerations, such networks need to be composed of essentially unreliable components. Traditional, passive fault tolerance solutions (e.g. duplex or triple-modular-redundancy) are often not viable in this context as they are expensive (by more than doubling the required resources) and do not facilitate keeping track of the system health.

One way to cope with the faults is to allow devices to fail and compensate failures at higher levels of the system stack [3] tolerating faults across layers involving hardware, firmware, operating system, applications etc. Cross-layer fault-tolerant systems have a potential to implement reliable, high-performance and energy-efficient solutions without an excessive cost [4] by distributing the fault tolerance tasks to multiple layers [5].

One of the promising concepts in fault tolerance is fault resilience, defined as the persistence of service delivery that can justifiably be trusted, when facing changes [6]. Resilience takes advantage of distributed systems, which provide inherent redundancy in the network. As a further step, the concept of cross-layer resilience, combining the cross-layer approach with fault resilience, has been proposed as a cost-effective and efficient solution to contend the faults [7]. According to Cheng et al. cross-layer resilience is a cooperation of multiple techniques from different layers of the system stack to achieve cost-effective error resilience, as a possible solution for designing low-cost resilient systems [8].

In sensor networks previous works on resilience have focused almost exclusively on wireless sensor networks utilizing network clustering and routing algorithms [9], [10] as well as adaptable network deployment [11].

Several frameworks for cross-layer resilience have been proposed. ERSA [12] addresses multi-/many-core systems, assuming one reliable core and several unreliable ones. However, it is tailored towards a special class of probabilistic applications utilizing the algorithmic level cognitive resilience. The framework called CLEAR is reported to deliver the targeted degree of resilience at low cost in terms of energy, power consumption, execution time and area by combining different resilience techniques across various layers of the system stack (circuit, architecture, OS, application) [8]. Both of the methods target processor cores and not the network.

IMMORTAL action [13] proposed cross-layer fault resilience and system health management for on-chip networks. A cross-layer fault-resilient Network-on-Chip (NoC) architecture Bonfire [14] , which includes concurrent checker circuitry, fault classification and cross-layer health management, was developed as a result [15]. Further, [16] developed a Phoenix distributed fault-tolerant architecture over an NoC-based multiprocessor platform that consists of a hardware part placed on each router of the NoC and a software part placed on the operating system of each processing element node composed of a paired processor-memory. The common denominator of the mentioned works is that they are constrained to single chip architectures. Moreover, most of the cross-layer resilient approaches (e.g. [13], [15] and [16]) require the addition of dedicated hardware for fault monitoring within the system. Other cross-layer fault tolerance approaches [3], [7], [17], [18] are implementing multiple layers of the system stack for resilience tasks. Multi-tier industrial scale fault detection and diagnosis has been described in [19], where tiers consist of different systems.

Data-driven and signal processing based fault detection and diagnosis is a concept [19], [20] that uses signal processing functions for fault diagnosis. Data-driven techniques may also allow detecting anomalies and faults that are not properly estimated by model-based fault detection [21]. Signal processing based fault detection is based on analysis of the output signals and does not always need explicit input-output model [22], [23] of the system. However, no hierarchy of data layers have been considered in the above-mentioned works.

This paper takes a different approach, where fault detection is data-driven cross-layer. Proposed approach does not focus on the system stack but rather on the data health of the sensor network. The main purpose of a sensor network is to measure, gather and deliver valid data. While the root cause of invalid data lies somewhere in the system stack, the concept of data-driven cross-layer resilience is based mainly on the purpose of the sensor network and less on the implementation details. In the approach proposed in the current paper the data layers are part of the sensor network's essential functionality itself and no additional dedicated hardware is needed. In proposed architecture data are collected exclusively by the functional sensors (the raw data layer) that are part of the system. Further data layers include the processed data and aggregated data from different sensor sources. As it will be shown in this paper, the approach allows detecting faults on a higher data layers even if they escape lower layer detection, and provides online diagnosis capabilities to the network in order to pin point the root cause and location of the error.

The main contribution of this work is proposing a combined data-driven and cross-layer resilient architecture for sensor networks, where instead of system stack layers, data layers are applied in fault detection and diagnosis. The data-driven approach utilizes the inherent redundancy within the sensor network as opposed to applying dedicated error monitors and/or duplicated hardware resources. The proposed approach enables keeping track of the system health within the sensor network in real time by using hierarchical health-maps. The proposed architecture also allows triggering actions like network reconfiguration and predictive maintenance to cope with failures in parts of the network. Ultimately, the proposed solution enables graceful degradation to sensor networks allowing them to continue their operation with degraded quality of service but in an uninterrupted manner even if a subset of the resources have failed.

In order to validate the proposed architecture, a case-study on an underwater sensor network implementation for flow monitoring in harbors has been carried out. The paper presents examples of cross-layer data detection during real incidents happening in the field. Moreover, synthetic experiments based on real in-field data demonstrate the effectiveness of the approach in fault detection and diagnosis. The experiments show that the data-driven cross-layer fault management allows improving the sensor group measurement accuracy by 35% in case of single sensor errors and nearly twofold in case of double sensor errors.

The rest of the paper is organized as follows. Section II presents developed software framework for managing sensor networks. Section III explains the data-driven cross-layer architecture for system health management. Section IV provides the real-world practical case study of an underwater sensor network for harbor water flow monitoring utilizing the proposed architecture and V presents an evaluation of the data-driven cross-layer fault detection and diagnosis in the field. In Section VI we discuss the limitations of proposed approach. Finally, Section VII concludes the paper.

II. SOFTWARE FRAMEWORK FOR SENSOR NETWORKS

For the purpose of managing sensor networks we have developed a generic software framework. The framework architecture is shown in Fig 1. In the following, a brief overview of the developed components divided into logical Edge, Fog and Cloud [24] computing environments is provided.

1) The Edge computing environment includes *Sensor Nodes* that measure environmental data and transfer it to a Sensor Gateway.
2) The Fog computing environment consists of a *Sensor Gateway* that is connected to multiple sensor nodes that may form *Sensor Groups*. A Sensor Gateway gathers Sensor data, processes it and publishes data further into the Broker in the Cloud Layer.
3) The Cloud computing environment has a *Broker* component which acts as a communication bus between the Cloud and Fog layers as well as between functional components in the Cloud layer. The term Broker originates from the MQTT [25] standard, however its functionality is not vendor-locked. The Cloud layer can also include an *Aggregator* that manipulates the data according to defined domain specific rules and publishes the manipulated (aggregated) data back to the Broker. The *Recorder* is subscribed to the Broker and listens to incoming data. It does not manipulate data in any way and is only recording it using predefined rules. *Subscribers* can subscribe to the Broker to interested data. This allows accessing different data feeds that pass the Broker with small latency. The *Web component* is presenting data from the Storage to End Users. The name is an abstraction that does not have to be a Web application but can be any kind of user interface.

III. DATA-DRIVEN CROSS-LAYER SYSTEM HEALTH MANAGEMENT ARCHITECTURE

A. Data layers

While previously (see Section II) we introduced the sensor network software components by computing environments, in this section we look at the sensor network from the

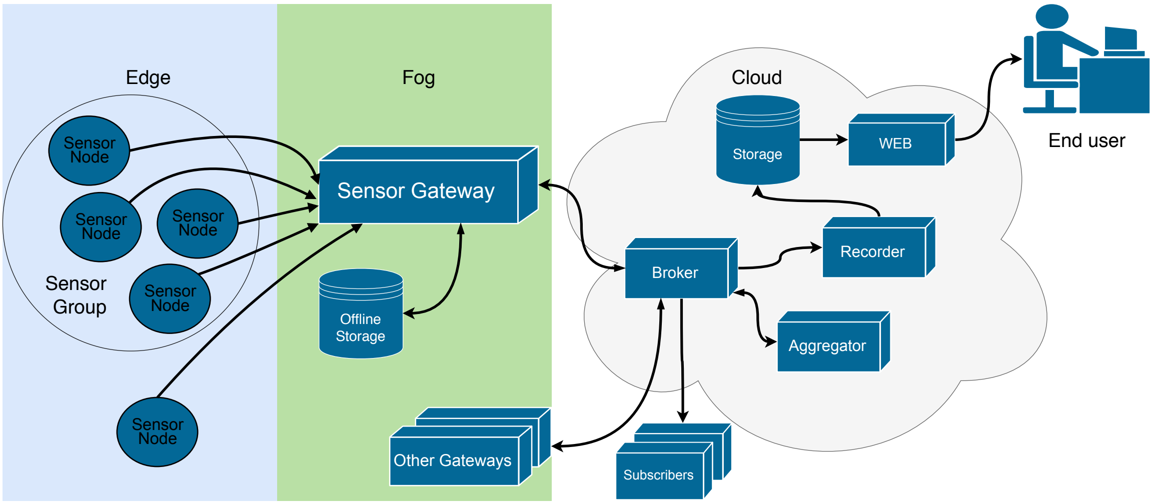


Fig. 1. Software architecture of the framework for sensor networks

perspective of the sensor data. Table I presents the data-driven layers of the sensor network architecture. There are three data layers - raw, processed and aggregated that correspond to different functionalities that can be loosely connected to given computing environments as shown in Table I.

At the *Raw data* layer (i.e. layer L_1), there is data that is measured and transferred by the set of Sensor Nodes $S = \{s_i\}$. The Raw data layer holds Sensor Node data before transformations and calibrations.

Processed data (layer L_2) is generated by the Sensor Gateway(s) (see Section II) from Raw Data. The processing step is essential in the current application (see Section IV), where sensor acceleration has to be transformed into the frequency domain. Additionally, during the processing outlier data is smoothed out and it is determined whether the data is within the measurement limits.

Aggregated data (layer L_3) is generated for the Sensor Groups $G = \{g_j\}$ by the Aggregator component (see Section II) from the Processed Data of the Sensor Nodes $\{s_i\}$. The purpose of the Aggregating process is to provide transition from single Sensor Nodes' data to Sensor Groups' data.

TABLE I
DATA LAYERS AND THEIR MAPPING TO COMPUTING ENVIRONMENTS

Data Layer	**State**	**Computing environment**
L_1: Raw	direct data from sensors	Edge
L_2: Processed	pre-processed data from sensors	Edge/Fog
L_3: Aggregated	data combined to sensor groups	Cloud

More formally, the data processing and aggregation takes place as follows. Let us assume sensors measuring and sending values periodically and the event of receiving a valid test packet at the Raw Data Layer L_1 by sensor s_i at a discrete time instance t be $\rho(i,t)$. All the valid $\rho(i,t)$ measurement values received during a time interval T, containing n time instances, are pushed into the raw data queue of length n, which acts as a First-In-First-Out (FIFO) data buffer:

$$Q = \{q_1, q_2, ..., q_n\},$$

where $q_1, ..., q_n$ represent n latest valid $\rho(i,t)$ measurement values.

To obtain the processed data at layer L_2, the data in Q are processed by a signal processing function Ψ to obtain the processed data value of $\sigma(i,t)$, i.e.

$$\sigma(i,t) = \Psi(Q).$$

Thereafter, the processed data value $\sigma(i,t)$ of the node s_i is passed through a calibration function Φ_i, i.e. the calibrated processed value of sensor s_i at time t is

$$\gamma(i,t) = \Phi_i(\sigma(i,t)).$$

Subsequently, the aggregated value $\alpha(j,t)$ of the sensor group g_j at layer L_3 is calculated by an aggregation function Ω on calibrated values $\gamma(i,t)$ of the sensor nodes s_i that belong to the sensor group g_j, i.e.

$$\alpha(j,t) = \Omega(\{\gamma(i,t) | s_i \in g_j\}).$$

Finally, the aggregated value $\alpha(j,t)$ is converted to the physical quantity $\beta(j,t)$ required by the end user by applying the conversion function Ξ:

$$\beta(j,t) = \Xi(\alpha(j,t)).$$

B. *Health-map hierarchy*

In the proposed architecture the current state of the system is collected into multilevel distributed Health-Maps. These Health-Maps are in essence hierarchical continuously updating data collections containing information about the health of the underlying data layer (e.g. successful/failed data packets during the latest interval). In the proposed architecture, we have defined a hierarchy of Health-maps at two levels: Node-Level Health-Maps $H_{Node}(i, t)$ for keeping track on the health status at Sensor Nodes s_i and Group-Level Health-Maps $H_{Group}(j, t)$ for the status of the Sensor Groups g_j at a discrete time instance t.

C. Fault detection and diagnosis at different data layers

This paper concentrates on hardware faults [26]) and how they are reflected on data that is generated, processed and aggregated at different data levels. In other words, model-free (i.e implicit) fault detection [20] is applied, i.e. erroneous data values at different layers are regarded as faults.

Using hierarchical health-maps (see Sect. III-B) and comparing gathered data to previously stored metrics, higher levels of data manipulation can isolate and smooth out the occurring faults.

In the following, description of the proposed fault detection mechanisms at different data layers and description of the cross-layer fault diagnosis is presented.

Faults at data layers L_1 and L_2 are detected and located with a resolution of a single sensor s_i. Let us denote the case of a fault being detected in sensor s_i at discrete time t at the Raw Layer L_1 by $f_1(i, t)$ and at the Processed Layer L_2 by $f_2(i, t)$, respectively. Denote by $\neg f_1(i, t)$ the case of no faults detected in sensor s_i at L_1 and $\neg f_2(i, t)$ no faults at L_2.

At every sensor s_i, one of the three cases presented below may occur:

1) $\neg f_1(i, t) \wedge \neg f_2(i, t)$ - i.e. the case, where no faults were detected at s_i;
2) $\neg f_1(i, t) \wedge f_2(i, t)$ - sensor s_i is still transmitting structurally valid packets. However, the data is erroneous and the fault is detected at L_2; This case corresponds to the case, e.g. where the sensor node s_i encounters a mechanical damage;
3) $f_1(i, t) \wedge f_2(i, t)$ - The node s_i has a complete electronic failure or the connection is broken.

Detection of faults at the aggregated data layer L_3 at time t occurs at sensor groups g_j and is denoted by $f_3(j, t)$. A sensor group is considered faulty if a critical number of sensors in that group are faulty. (For example, in the current implementation, a sensor group is regarded to be not faulty if at least 2 out of 4 nodes belonging to the group are still operational).

IV. CASE STUDY: UNDERWATER SENSOR NETWORK FOR HARBOR WATER FLOW MONITORING

A. Sensor Network Installation

We evaluated the proposed approach on an underwater network for monitoring sea currents in the harbor. The Sensor Nodes are installed to the underwater harbor infrastructure to notify approaching ships about the water flow around the piers. The goal is that berthing ships get information about the flow and turbulence from Sensor Nodes installed on the pillars of the pier. The Sensor Nodes are connected to the Sensor Gateway with underwater cables over RS-485 serial communication, thus the configuration of the underwater sensor network is fixed.

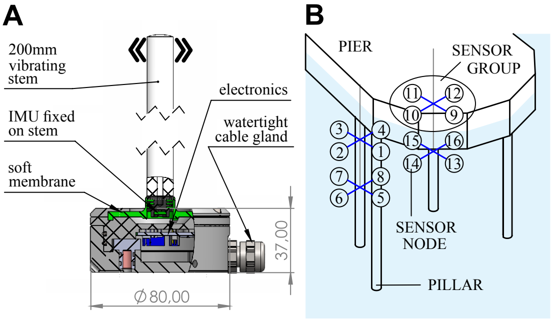


Fig. 2. A) Sensor Node B) Harbor Installation

The Sensor Node applied in the case study is shown in Fig. 2.A and its detailed description is given in [27]. The Sensor Nodes measure the flow magnitude and direction from the stem vibrating in the flow. An IMU (Inertial Measurement Unit) embedded into the Sensor Node calculates the accelerations of the stem in x and y directions. In the current implementation, to estimate the flow magnitude from the IMU data, a 15s series of IMU raw data is transformed with FFT (Fast Fourier Transform) into a frequency domain in 120s intervals and the PSD (Power Spectral Density) is used to find flow direction and magnitude using calibrations described in [27]. All the Sensor Nodes are wired to the Sensor Gateway in a star network topology.

The Sensor Nodes are installed around the pillars at two different depths so that at both depths 4 Sensor Nodes are attached around the pillar at 90 degrees angle from each other forming a logical Sensor Group. This is necessary because depending on the flow direction, the pillar itself always shelters some of the Sensors Nodes from the flow. Therefore each aggregation of 4 Sensor Nodes (Sensor Group) is used to estimate the flow at each point. Note however that the values of the Sensor Nodes in the same Sensor Group are correlated. Knowing how the flow should behave around obstacles helps us identify failing Sensor Nodes as well as to estimate how the readings of the sensors should be correlated.

In total the installation has 16 Sensor Nodes, grouped into 4 aggregations (Sensor Groups) of 4 sensors as shown in Fig. 2.B.

B. Layered data model implementation

For the example sensor network, the data layers described in Section III-A are specified as follows.

Raw data layer L_1: The raw measurement value $\rho(i, t)$ of the sensor node s_i at time t is a vector containing values such as Inertial Measurement Unit (IMU) XY positions, temperature and pressure.

Processed data layer L_2: The processing function Ψ first transforms the raw values $\rho(i, t)$ into a intermediate processed value $\sigma'(i,t)$, which is a vector containing median values of the IMU XY positions, of temperature and of pressure over all the elements in the processing data queue Q.

In addition, Ψ calculates the dominant frequency ($\sigma'.dfq$) by performing Fast Fourier Transform (FFT) over Q, taking half spectrum, finding index of maximum value, generating frequency scale and applying found index on frequency scale, which takes place at 120 second intervals. The $\sigma'.dfq$ value is obtained through a high pass filter in frequency domain by filtering out frequencies lower than 3Hz. The maximum value of the resulting array is assigned to this variable. $\Psi(Q)$ also provides the maximum value of Power Spectrum Density (PSD) $\sigma'.psd$ obtained from the FFT, where PSD values lower than 1000 are also filtered out.

Thereafter, $\Psi(Q)$ pushes the intermediate processed data value $\sigma'(i,t)$ to a processed data queue $P = \{p_1, p_2, ..., p_m\}$, where $p_1, ..., p_m$ hold the m latest intermediate processed data values. The processed data value $\sigma(i,t)$ is obtained as the median value of the elements in P, i.e. $\sigma(i,t) = median(P)$.

Aggregation layer L_3: As explained above, in the current implementation each sensor group g_j consists of 4 nodes s_i. The median value of the calibrated values $\gamma(i,t)$ of the 4 sensor nodes s_i becomes the aggregated value $\alpha(j,t)$. Finally, the conversion function Ξ converts $\alpha(j,t)$ into flow direction and velocity which are the physical quantity values $\beta(j,t)$ required by the end user of the harbor water flow monitoring application.

C. Health-map implementation and fault detection

In the presented case-study, the *Node Health-Map* $H_{Node}(i,t)$ is managed by the Sensor Gateway and contains information about each connected Sensor Node's traffic during the latest interval T. It contains counters of valid and invalid data packets as well as detected faults and a log for debugging purposes. In our implementation RS-485 serial communication was used and no automatic cyclic redundancy checking was implemented initially, thereby packet validation was done by the Sensor Gateway indirectly. At regular time intervals, in current implementation every 180s interval, this Health-map is published to the Broker for the Aggregator (see Section II).

More precisely, in the current implementation, the Node Health Map $H_{Node}(i,t)$ is defined as:

$$H_{Node}(i,t) = \{v_+(i), v_-(i), t_{ok}, \rho(i, t_{ok}), \Delta\sigma\},$$

where $v_+(i)$ is the number of valid packet reads and v_- is the number of invalid packet reads occurring in the sensor node s_i during T, respectively. t_{ok} is the time of the last good sensor measurement, $\rho(i, t_{ok})$ is the last known good sensor measurement value and $\Delta\sigma$ is the range of the dominant frequency values in the processed data queue P (See Sections III.A and III.B)).

A *Group Health-Map* $H_{Group}(j,t)$ keeps track of the healthy sensor nodes belonging to the sensor group g_j and it is managed by t he Aggregator component. The Aggregator is subscribed to the Broker for Node Health-Maps and in case of deteriorated health of any Sensor Node s_i, removes that node from its Sensor Group g_j in the Group Health-Map $H_{Group}(j,t)$.

In the current implementation, fault detection at the different data layers takes place as follows.

Fault detection at Raw Data Layer L_1: Invalid packets at Sensor nodes s_i (e.g. malformed or missing packets) at time instance t are filtered at this layer. In addition, the ratio of good and bad packets is being accumulated into the Node Health-Map $H_{Node}(i,t)$. At this level, problems that may occur and cause invalid packets are as follows:

- Data Decoding Error - the data packet does not correspond to the correct length or byte values and cannot be read from current stream.
- Redundancy check failing – One of the values in the data packet is the name of the Sensor Node, which should stay constant and never change.

The amount of deterioration of sensor s_i during T is characterized by the node deterioration rate $d_{Node}(i) = v_-(i) - v_+(i)$, which shows the difference on invalid and valid packets. If $v_+(i) = 0$ then the sensor s_i is fully deteriorated at Raw Data Layer and if $t_0 - t_{ok} < \Delta t$, where t_0 is current time and Δt is a predefined timeout period taking into consideration current environmental change rate, then $\rho(i, t_{ok})$ is used instead of the missing measured values. Otherwise, if the Δt is exceeded then the sensor is marked as currently faulty at Raw Data Layer, i.e. $f_1(i,t) = true$.

Fault detection at Processed Data Layer L_2: This layer includes an incoming FIFO buffer Q for the data from the raw data layer of which a snapshot is taken at a predefined interval T. The contents of Q are processed and results sent upstream. The processing steps at this data layer include the following.

- Buffer underflow – The FIFO buffer is cleared after every data acquisition by the processed data layer. If not enough good packets from the raw data layer came in during the last acquisition interval, the buffer may be not full. This indicates that not enough valid incoming data has been read from the sensor node.
- High pass filtering of data – Filtering out data values outside of measurement boundaries.

The range of the dominant frequency values $\Delta\sigma$ in the processed data queue P is written to the Node Health-Map $H_{Node}(i,t)$. If $\Delta\sigma = 0$ then a fault is said to be detected at the processed data layer L_2 (i.e. $f_2(i,t) = true$).

Fault detection at Aggregated Data Layer L_3: At this layer, the sensor groups are managed, single Sensor Nodes' data and Health-maps are analyzed, combined and aggregated. When a sensor fails, it is removed from its Sensor Group. If a critical number of sensor nodes have decayed then the group data cannot be trusted anymore and the sensor Group g_j is regarded as faulty.

Data aggregation in the current underwater sensor network takes place as follows. Let W_j be the set of sensor nodes s_i that are operational in group g_j, i.e. for them $\neg f_1(i,t) \wedge \neg f_2(i,t) = true$. In current implementation, the aggregated value $\alpha(j,t)$ of the sensor group g_j is the median value of calibrated values $\gamma(i,t)$ of sensor nodes $s_i \in W_j$. A group deterioration rate $d_{Group}(j)$ is calculated based on the functioning sensor nodes in it as $d_{Group}(j) = 1 - \frac{|W_j|}{|S_{g_j}|}$. In the current implementation, each sensor group consists of 4 nodes and at most 2 faulty sensor nodes per group (i.e. $d_{Group}(j) \leq 1/2$) are tolerated. If $d_{Group}(j) > 1/2$ then a group level fault is marked to be detected and $f_3(j,t) = true$.

V. EXPERIMENTAL STUDY AND FIELD OBSERVATIONS

In this Section we will present the results of synthetic scenarios based on real data showing the benefits provided by the cross-layer data-driven fault management approach. In addition, we will give some real-world examples of naturally occurred incidents, where faults were detected and diagnosed by the approach.

A. *Graceful data degradation enabled by fault management*

In current subsection, we are using real data recorded from natural measurements during a selected time interval. The faults are injected to this data by overriding data values in specific sensors by constant zero vectors. Thereby, synthetic fault scenarios are created. Then this set of data values is simulated to obtain the aggregated values for the sensor group.

Fig. 3 shows the data degradation in synthetic scenarios based on real data, where sensors were degraded. The comparison (see Fig. 3) includes proportional differences of the aggregated data calculated using failing sensors compared to the good aggregated data scenario with operational sensors. Proportional differences are found by using $max(x,y)/min(x,y)$ where x is the best scenario measurement value and y is correspondingly some other scenario measurement value at the same time. The orange line represents the case, where one faulty sensor has been removed from the aggregation of the group value. This is equal to the good aggregated value. The blue line represents the case, where one sensor is faulty (having $\sigma.dfq = 0$) but is still included to aggregated data computation. The red line represents scenario where 2 sensors are faulty (with $\sigma.dfq = 0$). Finally, the green line is showing the scenario where the 2 faulting sensors are removed from aggregated data computation by the fault manager. The experiments show that the data-driven cross-layer fault management allows in average improving the sensor group measurement accuracy by 35% (1.0 × versus 1.35 × deviation from the good value) in case of single sensor errors and nearly twofold (1.72 versus 3.18 × deviation) in case of double sensor errors (See the average values for different scenarios in Fig. 3).

In the following, we will show real data (without any overriding) from two different naturally occurred incidents, where fault detection was performed from data at different layers.

B. *Incident 1: Fault detection at the Raw Layer*

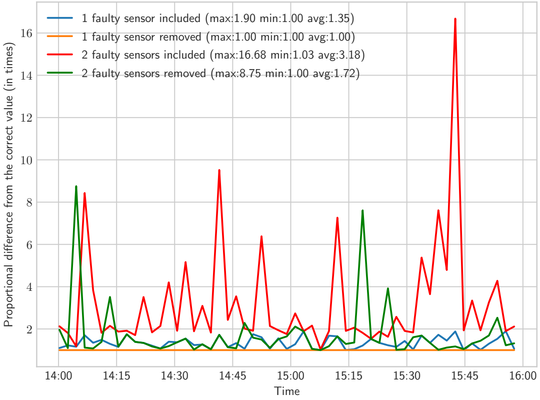


Fig. 3. Proportional difference in aggregated data

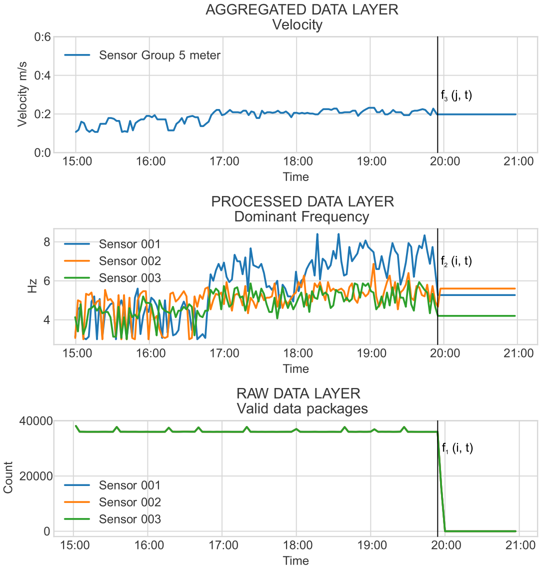


Fig. 4. Incident 1 time series: Fault manifestation at the Data Layers

Incident 1 illustrates a naturally occurred incident, which resulted in cables connecting the Sensor Nodes to the Sensor Gateway torn. Fig. 4 shows data from one Sensor Group at different Data Layers during a 6 hour interval. The Aggregated data layer shows median water flow velocity over all the nodes from a Sensor Group at 5 meter depth in the given time period. The Processed Data Layer in Fig. 4 shows Dominant Frequency on Sensors belonging to the same Sensor Group. The Raw Layer in Fig. 4 shows the packet count of valid measurements during 180s intervals during this period. On the raw data layer only one color can be seen as all the drawn nodes follow the same line. Although Sensor Groups consist of 4 Sensor Nodes, Node 004 was broken before the time of this event and it was removed from the Sensor Group. Therefore, its measurements are not drawn. In Fig. 4 all the layers clearly present an incident between 19:54-19:57, where correspondingly $f_1(i,t)$, $f_2(i,t)$ and $f_3(j,t)$ happened. The incident stopped incoming data from the Raw Layer that is represented as the number of incoming raw data packets dropping to zero. Processed and Aggregated Layers are showing last good values remaining constant because no new data was coming in after the incident. During the incident, connection cables were physically cut (See Fig. 5).

During the 6 hour period that is given around Incident 1, also the events from the Node level Health-map are quantified and shown in Table II. The few fault events recorded were from the incident where packets were disrupted ending up with bad packet length and decoding errors. However, this error quantity is low and can be a part of normal operation. It can be seen that for this type of incident, fault detection can be done from the Raw data layer monitoring the valid packet flow as shown in the lower section of Fig. 4.

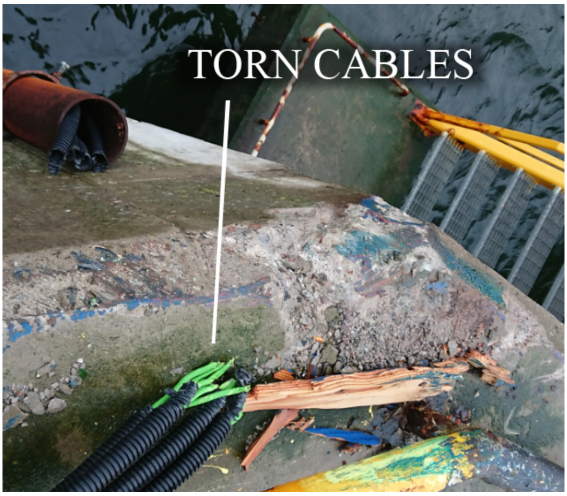


Fig. 5. Incident 1: Incident damage

TABLE II
INCIDENT 1 EVENT QUANTIFICATION (SAME INTERVAL AS FIG. 4)

Event type	**Count** **Sensor 001**	**Sensor 002**	**Sensor 003**
Valid packets	3559807	3559774	3559793
Data not yet ready	0	0	0
Bad packet length	3	2	2
Decoding error	0	1	0
Serial number changed	0	0	0
Port closed	0	0	0

C. Incident 2: Fault detection at the Processed Layer

Incident 2 represents an incident where a stem of a Sensor Node was detached. Fig. 6 shows data from different data layers of one node during a 1-hour interval. In this case it can be seen that valid packets kept coming in, however the measurements dropped to zero. Thus $f_2(i,t)$ occurred, but no $f_1(i,t)$. It was later found that the moving part of a sensor's stem was torn off as shown in Fig. 7. It can be seen from Fig. 6 that this type of incident was not identifiable from the Raw data layer, but rather from the sudden drop of the dominant frequency to zero at the Processed layer.

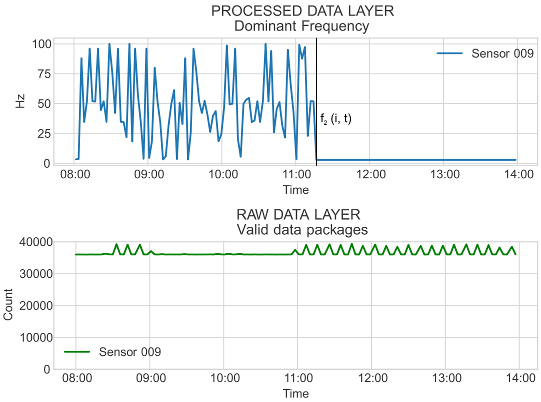


Fig. 6. Incident 2 time series: Fault manifestation at the Data Layers

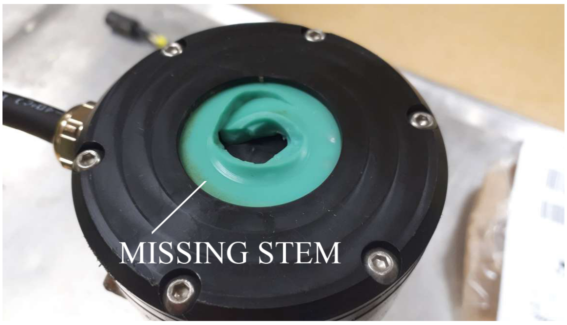


Fig. 7. Incident 2: Incident damage

Table III shows that there were no detected faults at the Raw layer ($\neg f_1(i,t)$) and the valid data packets count was higher than in Incident 1 (see Table II) due to the fact that the data flow was never interrupted.

TABLE III
INCIDENT 2 EVENT QUANTIFICATION (SAME INTERVAL AS FIG. 6)

Event type	**Count** **Sensor 009**
Valid packets	4317807
Data not yet ready	0
Bad packet length	0
Decoding error	0
Serial number changed	0
Port closed	0

VI. VALIDITY ASSESSMENT AND LIMITATIONS

The proposed architecture was implemented and tested in real life on a single sensor network. While our intention was to develop a generic sensor network architecture and it showed benefits in implemented water velocity measuring sensor network, it has not been validated on other type of real sensor networks.

The used sensor network for testing has essential signal processing needed for its functionality, so the data layers were not artificial and data-driven cross-layer fault management was integral addition to the core functionality of the sensor network. Not in case of all types of sensor networks the proposed architecture may be possible to implement or be resource and cost efficient.

VII. CONCLUSIONS

The paper proposed a data-driven cross-layer resilient architecture for sensor networks that combines fault detection across data layers into a coordinated system health management architecture. A practical experiments on an underwater sensor network for harbor water flow monitoring application were presented. The synthetic scenarios as well as in-field, real-world experiments for evaluating the cross-layer fault management capabilities and the aggregation of a system health-map based on the faults manifesting at the different layers were provided.

The incidents and synthetic scenarios based on real data show that the data-driven cross-layer fault management allows improving the sensor group measurement accuracy by 35% in case of single sensor errors and nearly twofold in case of double sensor errors. Additionally the proposed architecture is cost and resource effective as it relies on the sensor network internal functionality and no additional hardware is needed. The architecture is scalable in sense that there can be multiple hierarchies of sensor groups. In future the architecture can be developed further to dynamically manage sensor outliers and neighboring sensor group degradation. In addition we would like to implement and validate the same framework on different types of sensor network applications.

VIII. ACKNOWLEDGEMENTS

This research is funded by Estonian Centre of Excellence in ICT Research (EXCITE) and IT Academy Programme of Information Technology Foundation for Education.

REFERENCES

[1] D. Liu, "Efficient and Distributed Access Control for Sensor Networks," in *Handb. Secur. Cyber-Physical Crit. Infrastruct.*, pp. 227–250, Elsevier, 2012.

[2] S. Rehman, *Reliable Software for Unreliable Hardware – A Cross-Layer Approach*. Doctoral dissertation, Karlsruhe Institute of Technology (KIT), 2015.

[3] A. DeHon, H. M. Quinn, and N. P. Carter, "Vision for cross-layer optimization to address the dual challenges of energy and reliability," *Proc. -Design, Autom. Test Eur. DATE*, no. March, pp. 1017–1022, 2010.

[4] F. Campagnaro, R. Francescon, F. Guerra, F. Favaro, P. Casari, R. Diamant, and M. Zorzi, "The DESERT underwater framework v2: Improved capabilities and extension tools," in *2016 IEEE Third Underw. Commun. Netw. Conf.*, pp. 1–5, IEEE, aug 2016.

[5] M. Veleski, R. Kraemer, and M. Krstic, "An Overview of Cross-Layer Resilience Design Methods," in *Fifth Int. Conf. Radiat. Appl. Var. Fields Res.*, 2017.

[6] J.-C. Laprie, "From dependability to resilience," in *38th IEEE/IFIP Int. Conf. On Dependable Systems and Networks*, pp. G8–G9, 2008.

[7] S. Mitra, K. Brelsford, and P. N. Sanda, "Cross-layer resilience challenges: Metrics and optimization," *2010 Des. Autom. Test Eur. Conf. Exhib. (DATE 2010)*, no. March 2010, pp. 1029–1034, 2010.

[8] E. Cheng, S. Mirkhani, L. G. Szafaryn, C. Y. Cher, H. Cho, K. Skadron, M. R. Stan, K. Lilja, J. A. Abraham, P. Bose, and S. Mitra, "CLEAR: Crosslayer exploration for architecting resilience combining hardware and software techniques to tolerate soft errors in processor cores," *Proc. - Des. Autom. Conf.*, vol. 05-09-June, pp. 0–5, 2016.

[9] S. A. Sofi and R. N. Mir, "Natural algorithm based adaptive architecture for underwater wireless sensor networks," *Proc. 2017 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2017*, vol. 2018-Janua, pp. 2343–2346, 2018.

[10] S. Tyagi and N. Kumar, "A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 36, no. 2, pp. 623–645, 2013.

[11] C. H. Wu, K. C. Lee, and Y. C. Chung, "A Delaunay Triangulation based method for wireless sensor network deployment," *Comput. Commun.*, vol. 30, no. 14-15, pp. 2744–2752, 2007.

[12] L. Leem, H. Cho, J. Bau, Q. A. Jacobson, and S. Mitra, "ERSA: Error resilient system architecture for probabilistic applications," *Proc. -Design, Autom. Test Eur. DATE*, pp. 1560–1565, 2010.

[13] G. Aleksandrowicz, E. Arbel, R. Bloem, T. D. ter Braak, S. Devadze, G. Fey, M. Jenihhin, A. Jutman, H. G. Kerkhoff, R. Könighofer, S. Koyfman, J. Malburg, S. Moran, J. Raik, G. Rauwerda, H. Riener, F. Röck, K. Shibin, K. Sunesen, J. Wan, and Y. Zhao, "Designing Reliable Cyber-Physical Systems," in *Lang. Des. Methods, Tools Electron. Syst. Des. Lect. Notes Electr. Eng. vol 454* (F. Fummi and R. Wille, eds.), pp. 15–38, Springer, Cham, 2018.

[14] "Project Bonfire - network-on-chip." GitHub Repository, 2015.

[15] S. P. Azad, B. Niazmand, K. Janson, N. George, A. S. Oyeniran, T. Putkaradze, A. Kaur, J. Raik, G. Jervan, R. Ubar, and T. Hollstein, "From online fault detection to fault management in Network-on-Chips: A ground-up approach," in *2017 IEEE 20th Int. Symp. Des. Diagnostics Electron. Circuits Syst.*, pp. 48–53, IEEE, apr 2017.

[16] J. Silveira, C. Marcon, P. Cortez, G. Barroso, J. M. Ferreira, and R. Mota, "Scenario preprocessing approach for the reconfiguration of fault-tolerant NoC-based MPSoCs," *Microprocess. Microsyst.*, vol. 40, pp. 137–153, feb 2016.

[17] S. S. Sahoo, B. Veeravalli, and A. Kumar, "Cross-layer fault-tolerant design of real-time systems," in *2016 IEEE Int. Symp. Defect Fault Toler. VLSI Nanotechnol. Syst. DFT 2016*, pp. 63–68, IEEE, sep 2016.

[18] N. P. Carter, H. Naeimi, and D. S. Gardner, "Design techniques for cross-layer resilience," *2010 Des. Autom. Test Eur. Conf. Exhib. (DATE 2010)*, no. February, pp. 1023–1028, 2010.

[19] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Trans. Ind. Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013.

[20] S. X. Ding, *Data-driven Design of Fault Diagnosis and Fault-tolerant Control Systems*. Advances in Industrial Control, London: Springer London, 2014.

[21] P. Freeman, R. Pandita, N. Srivastava, and G. J. Balas, "Model-based and data-driven fault detection performance for a small UAV," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 4, pp. 1300–1309, 2013.

[22] A. Zhirabok and S. Pavlov, "Data-driven method of fault detection in technical systems," *Procedia Eng.*, vol. 100, no. January, pp. 242–248, 2015.

[23] R. Golombek, S. Wrede, M. Hanheide, and M. Heckmann, "Online data-driven fault detection for robotic systems," in *2011 IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, pp. 3011–3016, IEEE, sep 2011.

[24] J. Bar-Magen Numhauser, "Fog computing introduction to a new cloud evolution," *Univ. Alcalá*, 2012.

[25] "Information technology - Message Queuing Telemetry Transport (MQTT) v3.1.1," standard, International Organization for Standardization, Geneva, CH, June 2016.

[26] T. Wilfredo and W. Torres-Pomales, "Software Fault Tolerance: A Tutorial," Tech. Rep. October, NASA, 2000.

[27] A. Ristolainen, J. A. Tuhtan, and M. Kruusmaa, "Continuous, near-bed current velocity estimation using pressure and inertial sensing," *IEEE Sens. J.*, vol. PP, no. c, pp. 1–1, 2019.

Appendix 2

II

L. Vihman, M. Kruusmaa, and J. Raik. Systematic Review of Fault Tolerant Techniques in Underwater Sensor Networks. *Sensors*, 21(9):3264, may 2021

sensors

Review

Systematic Review of Fault Tolerant Techniques in Underwater Sensor Networks

Lauri Vihman [1,2,*], **Maarja Kruusmaa** [2] and **Jaan Raik** [1,2]

[1] Smart City Center of Excellence (Finest Twins), Tallinn University of Technology, 12618 Tallinn, Estonia; jaan.raik@taltech.ee
[2] Department of Computer Systems, Tallinn University of Technology, 12618 Tallinn, Estonia; maarja.kruusmaa@taltech.ee
* Correspondence: lauri.vihman@taltech.ee

Abstract: Sensor networks provide services to a broad range of applications ranging from intelligence service surveillance to weather forecasting. While most of the sensor networks are terrestrial, Underwater Sensor Networks (USN) are an emerging area. One of the unavoidable and increasing challenges for modern USN technology is tolerating faults, i.e., accepting that hardware is imperfect, and coping with it. Fault Tolerance tends to have more impact in underwater than in terrestrial environment as the latter is generally more forgiving. Moreover, reaching the malfunctioning devices for replacement and maintenance under water is harder and more costly. The current paper is the first to provide an overview of fault-tolerant, particularly cross-layer fault-tolerant, techniques in USNs. In the paper, we present a systematic survey of the techniques, introduce a taxonomy of the Fault Tolerance tasks, present a categorized list of articles, and list the open research issues within the area.

Keywords: underwater sensor network; fault tolerance; cross-layer fault tolerance; fault management

Citation: Vihman, L.; Kruusmaa, M.; Raik, J. Systematic Review of Fault Tolerant Techniques in Underwater Sensor Networks. *Sensors* **2021**, *21*, 3264. https://doi.org/10.3390/s21093264

Academic Editor: Yunyoung Nam

Received: 26 March 2021
Accepted: 2 May 2021
Published: 8 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Underwater Sensor Networks (USNs) have become widespread and are being deployed in a wide range of applications ranging from harbor security to monitoring underwater pipelines and fish farms. Due to the fact that USNs often operate in an extremely harsh environment, and many of their applications are safety-critical, it is imperative to develop techniques enabling these networks to tolerate faults. Moreover, USNs face many challenges that are not present in terrestrial networks, such as virtual inapplicability of the wireless radio communication under water and limitations of the acoustic means, for example.

In the current paper, applications, practices, and central issues on fault tolerant USNs are discussed, and a systematic survey of fault tolerant techniques in USN networks is presented. Our objective is to investigate the state of the art and main focuses of ongoing research on cross-layer Fault Tolerance in underwater sensor networks, as well as to identify the existing gaps in previous research. As by now a limited effort has been put on the Fault Tolerance of USNs by the research community, the criteria is expanded, and papers covering some specific aspects of the fault-tolerance topic are also taken into account. Moreover, the sources also include generic terrestrial Fault Tolerance in sensor networks because research on underwater sensor network faults is limited, and many of the generic technologies, approaches, and tools can be adapted for use in USNs.

It is important to stress that the underwater environment is mostly different from terrestrial conditions, in the sense of additional and more fatal hazards, like an increased pressure and a danger of flooding, as well as added difficulty of communication and physical access. Some communication media, such as radio signals, are not applicable underwater. Additionally, falling temperatures with increasing depth may affect the equipment's operation and reliability.

In this paper, a systematic search in IEEExplore, Google Scholar, and ScienceDirect online environments was carried out to obtain a relevant sample of works in the field of fault tolerant techniques in USNs. The search revealed 122 papers, with 59 of them dedicated to the Fault Tolerance of USNs, while the 63 remaining ones were generic fault tolerant techniques for terrestrial sensor networks applicable to the underwater environment.

In order to provide a systematic view of the paper categories, this survey introduces a taxonomy of Fault Tolerance tasks. Specifically, the identified relevant papers are grouped according to the tasks of fault prevention and prediction and Fault Detection and Fault Identification, as well as Fault Isolation and Fault Masking, respectively.

Moreover, a comparative analysis of the identified papers was presented, where the works were characterized according to their extra-functional aspects covered (i.e., security, energy-efficiency, scalability, cross-layer aspect) and Fault Tolerance tasks targeted, as well as marine or terrestrial application. As a result of the analysis, the lack of cross-layer Fault Tolerance approaches in the USN domain was identified as a particular gap in the state-of-the-art with prospective future research.

There are several surveys investigating underwater sensor networks. For example, Reference [1] introduced the term of Internet of Underwater Things (IoUT) and showed its applications in fish farms, monitoring underwater pipelines, harbor security, etc., and [2] analyzed cross-layer error control in Underwater Wireless Sensor Networks (UWSNs); however, the analysis focused on the underwater wireless network functionality faults and not on other sources of the USN faults. Underwater communications have been specifically surveyed in Reference [3,4], disregarding aspects of underwater sensor networks outside communication issues.

The main challenges identified for Internet of Underwater Things are the communication reliability and the differences between Underwater and Terrestrial Networks [5], such as mobility caused by water flow. For terrestrial sensor networks, there were 11 surveys found. Thereof, 3 terrestrial surveys addressed cross-layer aspects. Reference [6] was surveying cross-layer resilience design methods and [7,8] fault management techniques in wireless sensor networks. In addition, Reference [9,10] included surveys about aspects of the internet of things, and 7 papers by References [7,8,11–15] were surveys of different aspects of terrestrial wireless sensor networks. Reference [16] presented a survey about fault tolerant control systems, and, finally, Reference [14] was focused on surveying fault management frameworks in terrestrial wireless sensor networks.

The current state-of-the-art is lacking literature reviews covering faults in USNs not only from communication but from the entire infrastructure perspective, as well. To that end, the current paper has the following novel contributions:

- to the best of the authors' knowledge, this is the first survey of fault-tolerant, particularly cross-layer fault-tolerant, techniques in USNs;
- it introduces a taxonomy of the Fault Tolerance tasks for categorizing fault-tolerant techniques for USNs;
- it presents a comprehensive, categorized list of articles of works applicable in fault-tolerant USN design and deployment; and
- the survey also lists the open research issues within the focused area.

The paper is organized as follows. In Section 2, the formal methodology of selecting the papers is explained and a breakdown of the sample by keywords is provided. Section 3 gives an overview of the specific Fault Tolerance challenges in underwater sensor networks. In Section 4, the taxonomy of possible fault sources and that of Fault Tolerance tasks is presented. Subsequently, Section 5 is divided according to this taxonomy of tasks. In Section 5.1, works targeting the fault prevention and prediction task are discussed and the respective design, deployment, data collection, and testing frameworks are reviewed. Section 5.2 gives an overview of Fault Detection and Fault Identification techniques. Section 5.3 provides an overview of Fault Masking and Fault Recovery techniques. In Section 6, a categorized table of the related works identified by the survey is presented. Finally, in Section 7, open research issues are discussed, and conclusions are drawn in Section 8.

2. Methodology

The current overview is following the PRISMA [17] guidelines for systematic reviews. In order to obtain a relevant sample in the field of fault tolerant techniques in USNs, IEEExplore, Google Scholar, and ScienceDirect online environments were searched with the following search keywords: "underwater", "sensor network", "internet of things", "resilient", "fault tolerant", "fault management", "cross-layer" in English language. Because the resulting counts were low (see Table 1), some keywords were removed, and more papers identified. Top papers were selected by the order of relevance offered by the respective environments. The papers published before the year 1990 were not considered. Further, citations within those sources were searched from the aforementioned environments, and additional papers were identified this way. Related articles offered by IEEExplore and ScienceDirect algorithms were also taken into account. Next, the duplicates and non-relevant papers (e.g., control theory) were removed from the collected papers, and the collected papers were analyzed, categorized, and divided into marine and terrestrial categories. Personalization on search engines was turned off wherever possible.

Table 1 shows the count of results using combinations of keywords in Google Scholar, IEEExplore, and Sciencedirect. (Searches were conducted on 13 April 2021, from Taltech, Estonia, IP addresses.). From Table 1, it can be seen that some combinations were giving no, or a very limited number of, results. A critical amount of papers was not reached using the initial criteria, and the criteria were expanded to include also relevant non-marine-specific (terrestrial) papers. The argumentation behind this is that many of these techniques may also be usable in underwater environments (see Section 3).

Table 1. Search engine result count of respective keyword combinations.

Search Keywords	G. Scholar	IEEEX	S.Direct
"underwater", "internet of things", "resilient", "fault tolerant", "fault management", "cross-layer"	4	0	0
"underwater", "internet of things", "resilient", "fault tolerant", "fault management"	8	0	1
"underwater ", "sensor network", "resilient", "fault tolerant", "fault management"	36	0	4
"sensor network", "resilient", "fault tolerant", "fault management", "cross-layer"	49	1	4
"underwater ", "sensor network", "fault management"	162	0	10
"sensor network", "resilient", "fault tolerant", "fault management"	223	9	16

As a result of the search procedure, 122 related works were identified. These included 59 papers on marine Fault Tolerance and 63 papers being on terrestrial. The papers were tagged by specific areas addressed by them. The tags for specific areas included 'sensor network', 'fault tolerant', 'wireless', 'scalable', 'mobile', 'routing protocol', 'security', 'localization', 'framework', 'survey', 'energy-efficient', 'cross-layer', 'deployment', 'marine', and 'terrestrial'.

A bar graph showing the number of papers from our search that covered different specific areas is presented in Figure 1. The specific areas are ordered by the number of papers addressing them, and the bars for the specific areas maintain their colors throughout Figures 1–3. It should be noted that, in the following context, the meaning of "localization" is location detection in space, and the meaning of "mobile" is capacity of movement. It can be seen from Figure 1 that there were substantially more terrestrial papers than the ones specific to marine environments. In addition, wireless communication is a frequently targeted area. Figure 2 shows research areas of the analyzed papers falling into terrestrial category. It should also be noted that papers on general fault-tolerant sensor networks, not specifically claiming any environments, were categorized into the terrestrial category. Figure 2 presents the frequency of specific areas addressed in terrestrial papers where the order of the most frequent categories has switched but is not much different from Figure 1. However, Figure 3, which presents the analyzed marine and aquatic environment-related papers covering different specific areas, shows that marine wireless communication related research works have the highest number of papers among those identified by the current survey.

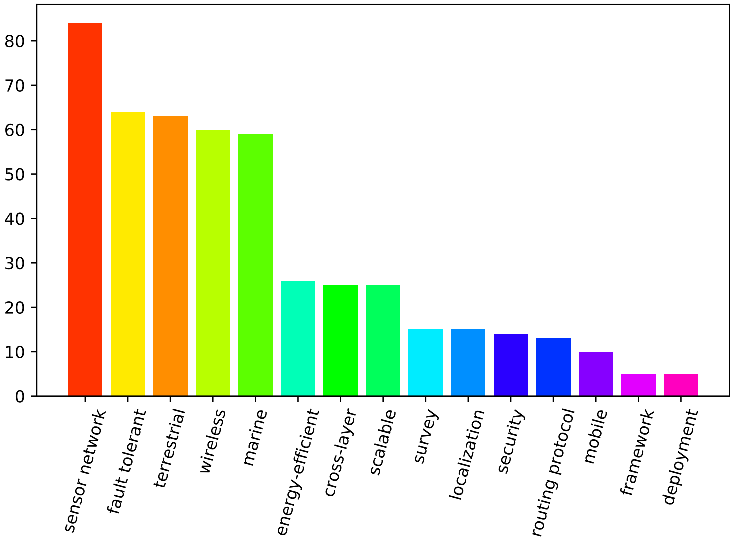


Figure 1. The number of papers by specific areas.

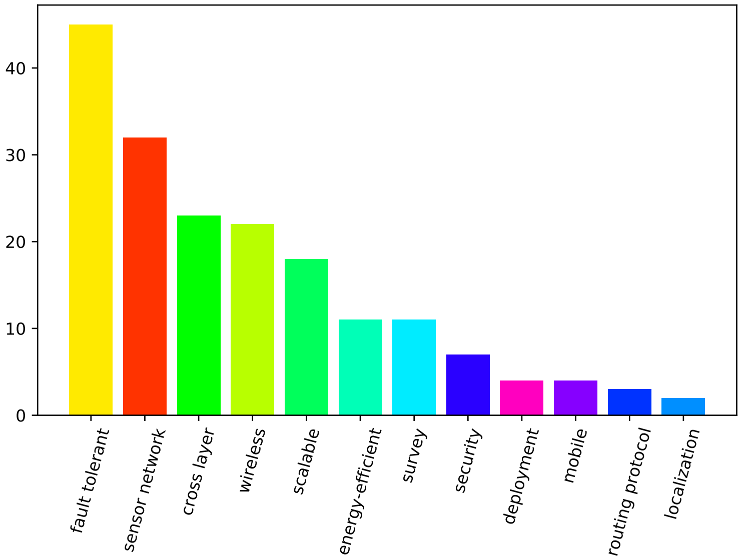


Figure 2. The number of terrestrial-related papers by specific areas.

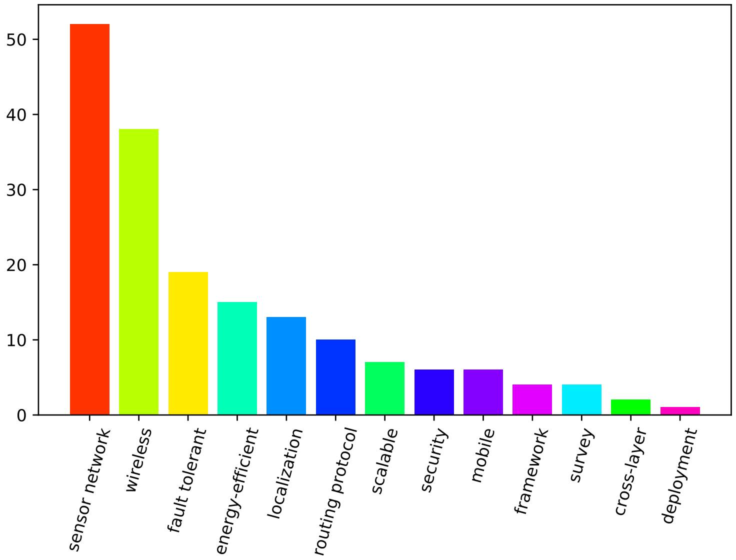


Figure 3. The number of marine-related papers by specific areas.

In order to further highlight the differences of the previous research focus in marine and terrestrial sensor networks, a radar diagram is shown in Figure 4. For the diagram, we selected eight significant specific areas: 'fault tolerant', 'wireless', 'mobile', 'localization', secure', 'scalable', 'energy efficient', and 'cross-layer', respectively. It can be seen from Figure 4 that a large share of marine research (shown by blue color in Figure 4) interest from the identified papers has been drawn to underwater wireless communication, while some are drawn to underwater Fault Tolerance techniques and almost none to underwater cross-layer Fault Tolerance. Underwater energy-efficiency and scalability are more covered areas than underwater vehicles (mobility) and security. Papers addressing terrestrial techniques (shown by green in Figure 4) were, according to the initial search criteria, more focusing on Fault Tolerance, including cross-layer Fault Tolerance, and less on energy efficiency or security.

High research effort on marine wireless networking in Figure 4 confirms the claim [5] that current pace of research on Internet of Underwater Things (IoUT) is slow due to the challenges arising from the uniqueness of underwater wireless sensor networks. Specifically, the main challenges for IoUT are the differences between Underwater Wireless Sensor Networks and Terrestrial Wireless Sensor Networks [5].

Fault Tolerant Control Systems is another extensive research area of Fault Tolerance not covered by current paper. There is an existing recent review paper [16] on the overview of research works in that topic.

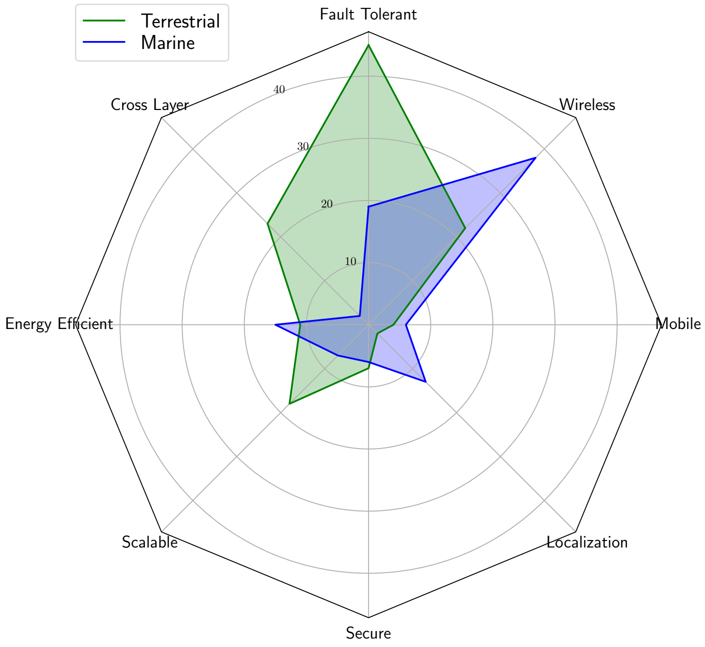


Figure 4. A radar chart of the analyzed papers addressing the main specific areas.

3. Specifics of Underwater Sensor Networks

Environmental and engineering challenges for sensor networks in underwater environments are shown on Figure 5. An underwater environment is mostly different from a terrestrial one due to the harsh physical conditions—high pressure and hard accessibility, as well as limited communication and energy resources. Depending on the specific location, the temperature may fall with increasing depth, which may affect, e.g., the battery lifetime. In underwater environments, faults can be caused over time by ambient flowing water generated by surface waves or other reasons that shake the components of the sensor networks. Moreover, faults can be introduced by humans or aquatic organisms.

Many communication methods are unavailable underwater, and there are multiple phenomena [2,18] that obstruct communication there. Because of the possibility of flooding the hardware due to water leakage, more attention and resources should be paid to the physical integrity of sensor nodes. On the other hand, faults from excessive heat should be rare and avoidable underwater. In the underwater context, Fault Tolerance has been so far addressed for reliant UWSN networking [2,3,19,20], space localization [21], and monitoring underwater pipelines [22]. While it should be possible to adapt most of the generic Fault Tolerance concepts for the underwater use, the environment is more demanding and unforgiving, and faults are more costly. Some more demanding approaches, like cloud computing, may not make sense to be implemented in USNs. However, the authors cannot see any obstacles for applying those fault tolerant approaches that yield appropriate communication methods, low network bandwidths, and power requirements in the underwater domain.

Last but not least, one of the promising approaches that could be adapted successfully within the underwater environment's constraints appears to be cross-layer resilience, which is an open research topic and lacking in recent research works, even for the terrestrial implementations.

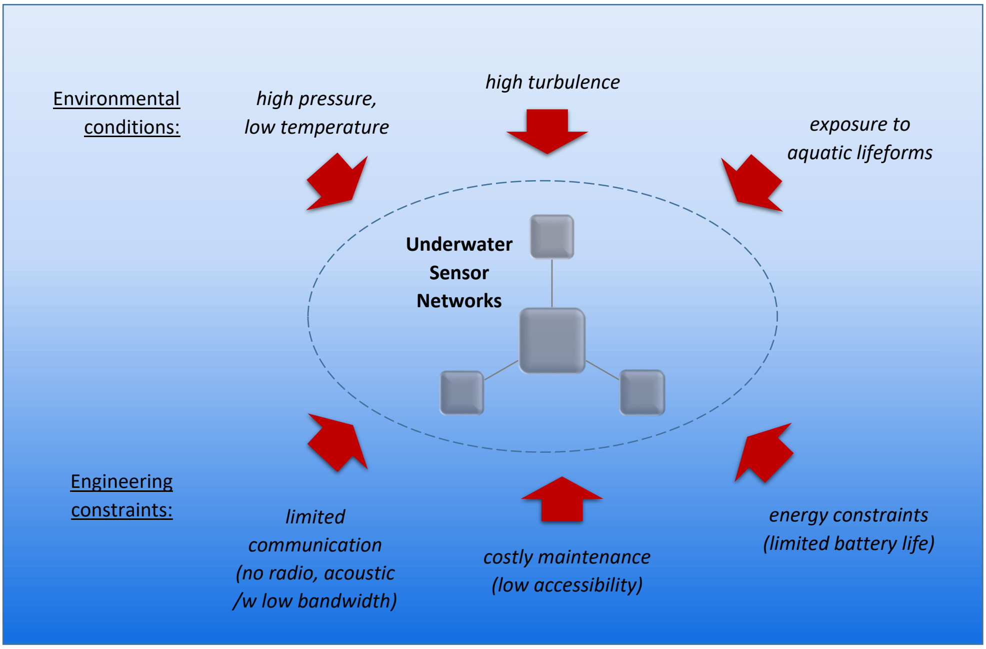

Environmental conditions:
high pressure, low temperature
high turbulence
exposure to aquatic lifeforms
Underwater Sensor Networks
Engineering constraints:
limited communication (no radio, acoustic /w low bandwidth)
costly maintenance (low accessibility)
energy constraints (limited battery life)

Figure 5. Environmental and engineering challenges in USNs.

4. Taxonomy of Faults and Fault Tolerance Tasks

In the following, we present the taxonomy of the sources of faults, as well as of the Fault Tolerance tasks. The objective of describing and representing these taxonomies is to categorize the articles for the current survey.

4.1. Sources of Faults

A fault is defined [23] as an underlying defect of a system that leads to an error. An error is a faulty system state, which may lead to failure, and failure is an error that affects system functionality. Faults may occur in different components and layers of systems for different reasons. The only type of fault possible in software is a design fault introduced during the software development, i.e., a bug [24]. Software bugs can be addressed separately and will not be covered further in the current paper.

Fault sources can be categorized by components where they occur. In sensor networks, they can occur in sensor nodes, in the communication network, and in the data sink [25]. Sensor networks share common failure issues with traditional networks, as well as introduce node failures as new fault sources [7].

USNs additionally introduce faults caused by environmental conditions, such as pressure, currents, underwater obstacles, etc. Those conditions may cause physical damage that may result in failures, as well as obstruct the system's functionality. For instance, in underwater acoustic networks, loss of connection and high bit error rate may be caused by shadow zones [18] formed by different physical reasons. Domingo and Vuran distinguish up to five different underwater propagation phenomena which may obstruct communication [2].

Faults can either be permanent or temporary [26]. Permanent faults may be caused by manufacturing defects, as variances of the hardware components are inevitable due to physical reasons [27]. One of the other factors that can introduce faults is aging and wear-out of the hardware components [28]. In addition to the components themselves, the interconnections between them are also affecting the reliability and may cause faults [29].

One of the challenges of fault management is temporary faults, especially soft errors. Soft error is a temporary change of signal value due to ionizing particles [26] that may lead to failure. Due to high integration density, it is estimated that soft failure rate is increasing in the future [30]. Another potential source of temporary faults is electromagnetic interference [31].

4.2. Fault Tolerance Tasks

The objective of the current section is to define a taxonomy of Fault Tolerance tasks to help categorize the identified papers. The Fault Tolerance tasks are based on more general Fault Tolerance principles from Reference [32,33]. Figure 6. shows the taxonomy of Fault Tolerance tasks applicable in USNs and how they affect each other. While the design and initial deployment of USNs contribute to Fault Prevention and Prediction abilities, data collecting techniques at the run-time contribute also to Fault Detection and Fault Recovery stages of the system, all of which are going to be discussed in the current paper.

The techniques under consideration can be categorized into the following groups:

- Fault Prediction and Prevention
  This task is about both preventing a fault to happen, as well as about proactive fault avoidance. Sensor networks can prevent certain faults from happening by design and/or deployment aspects. A disadvantage of fault prevention is a potentially increased system complexity. Fault avoidance, in turn, includes manufacturing testing and verification, which have a high cost often exceeding that of the entire design process.
- Fault Detection and Identification
  One of the central parts of Fault Tolerance is Fault Detection and Fault Identification of affected components which can, for instance, be performed by utilizing data collection with ping messages. Without Fault Identification, for instance, sensor node and network faults may be hard to distinguish. A disadvantage of Fault Detection and Fault Identification may be additional energy requirements and network congestion.
- Fault Isolation, Masking, and Recovery
  Isolation, masking, and recovery are different techniques for repairing a fault, minimizing the effect of a fault, or avoiding it to turn to system failure. Identified faults can be isolated, masked, and sensor network recovered, for instance, redirecting traffic through healthy backup components. Fault Recovery can ensure overall system operation even when components degrade. The downside may be the cost of adding components to ensure redundancy.

The overview of fault tolerant techniques presented in the following section follows the above-described taxonomy.

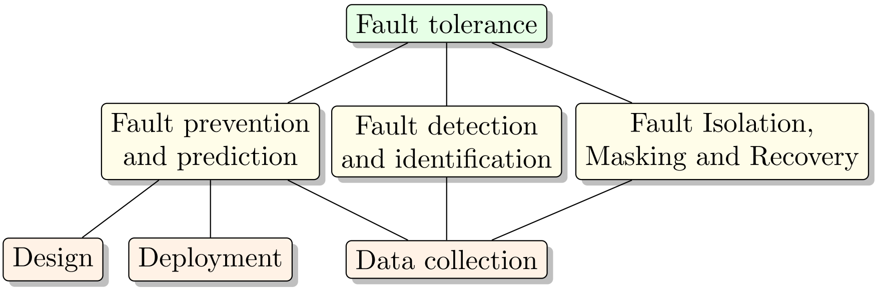

Fault tolerance
Fault prevention and prediction
Fault detection and identification
Fault Isolation, Masking and Recovery
Design
Deployment
Data collection

Figure 6. Taxonomy of Fault Tolerance tasks in USNs.

5. Overview of Techniques by Fault Tolerance Tasks

In the following, the Fault Tolerance techniques categorized according to the Fault Tolerance Tasks introduced in Section 4.2 and presented in Figure 6 will be discussed in more detail.

5.1. Fault Prevention and Prediction

Fault prevention and prediction in sensor networks are dependent on the architectural design of the system and the initial deployment method of the sensor network. These will be discussed in the following subsections. In addition, data collection in USNs and testing frameworks for UWSNs are presented.

5.1.1. Design of the Sensor Network

In Wireless Sensor Networks (WSN), instead of a centralized homogeneous topology, dividing nodes into clusters is an energy efficient and resilient method [12], where dedicated cluster head nodes may have more energy and communication capabilities to effectively act as mediators between regular nodes and data sinks.

To overcome the issues caused by varying environmental challenges of Underwater Wireless Sensor Networks (UWSN), natural algorithms may be utilized. For instance, clustering and routing can be done utilizing Cuckoo Search algorithm and Particle Swarm Optimization [34], which have behaved more resiliently in underwater conditions than more usual terrestrial Low Energy Adaptive Clustering Hierarchy (LEACH) protocol [11]. Pressure measurements have been used for UWSN routing [35] with floating depth-controlling sensors. Fault Management tasks can also be distributed across the whole network. In WSN with enough spare nodes energy efficient grid can be formed [36], changing the node manager, gateway and sensing nodes selected and spare nodes put to sleep. This results in energy-efficient and lightweight network but requires excess nodes.

However, existing UWSN protocols have not been adequately compared in underwater field trials yet [4].

5.1.2. Sensor Network Deployment

Sensor network deployment techniques are important for WSNs where deployment may directly affect the nodes' locations and networking availability. Even for terrestrial wireless sensor networks, to obtain a satisfactory network performance, an adaptable deployment method is essential [37]. Usually, the sensor placement for WSNs utilizes, for redundancy reasons, more sensors than the minimum required number [38]. The deployment costs and energy efficiency of WSNs have been investigated in Reference [39], and it has been found that there is no single solution that can easily be applied in practice [40].

Wired sensor network deployment is less researched, possibly because wired sensor networks' node deployment locations are limited by the cables, their locations are more predetermined, and node connectivity is not directly related to the location.

5.1.3. Data Collection

Sensor networks tend to have limited network bandwidth, energy, and storage capabilities. Thus, filtering and aggregating sensor information may be a way to meet those requirements. Raw sensor data near the source can be divided into informative, non-informative, and outlier groups [41], and only the needed data could be communicated or stored. Outlier data may result from noise, failures, disturbances, etc., and may be useful for Fault Tolerance purposes.

Different techniques to compress and aggregate collected information in UWSNs are investigated in Reference [42]. It was found that aggregation is justified, and cluster-based aggregation techniques are performing better than non-cluster-based ones. For instance, cluster head (CH) switching to backup (BCH) technique was proposed [43] for cluster-based UWSNs.

Moreover, security challenges need to be addressed. One way to minimize the risk of data tampering and/or interference is to ensure that the data is processed locally or, if that is not possible, then communicated end-to-end encrypted [44].

5.1.4. UWSN Testing Frameworks

Wireless networking protocols are one of the key research areas in UWSNs. To evaluate the implementation of underwater wireless protocols, simulation is often used. Due to the specifics of underwater environments (See Section 3), generic simulation environments are not able to capture some of the relevant features. Frameworks covered in the current section are useful for underwater acoustic protocols' simulation and evaluation.

Frameworks, such as DESERT version 1 and 2 [45] and SUNSET [46], that allow simulation, emulation, and testing of the sensor networks, have been developed for UWSNs. An analysis conducted in Reference [47] shows that SUNSET represents a more mature, flexible, and robust framework for in-field testing than DESERT. However, DESERT v2 was released subsequently. For acoustic UWSN security testing, SecFUN framework [48] has been proposed.

5.2. Fault Detection and Identification

In essence, Fault Detection means determining that one or more bits in the computation differ from their correct value [33]. This can be detected via continuous monitoring of the network and nodes' status. Some sources also use the word "Diagnosis" in a broader meaning than just detection and identification. Diagnosis has been defined as "characterizing the system's state to locate the causes of errors, determine how the system is changing over time, and predict errors before they occur [33]". The current section covers different techniques to execute the previously mentioned concepts.

A distributed hierarchical fault management [49] has been used for WSNs, where agent Fault Detection devices collect information from the power modules and sensors to determine failure conditions and sequentially diagnose the nature of the detected failure.

At higher abstraction levels, there has been a wide use of the SNMP protocol [50] by the industry for Fault Detection querying and triggering in IP networked devices. There are multiple commercial tools for generating failures, e.g., Chaos Monkey from Netflix [51], that randomly terminate services in production environments, to ensure their resiliency. The latter does not manage the occurring faults but ensures that the repairing mechanisms are in place and operable. Intelligent Platform Management Interface (IPMI) [52] is an industrial technology specification for hardware system management and monitoring.

A neural-network-based scheme for sensor failure detection, identification, and accommodation can be used which may allow the conditions to deviate to greater extent from theoretical models and estimation. A relatively simple and computationally light approach has been presented [53], where a neural network is used as an online learning state estimator for detecting faults. The neural network itself can be built as fault-tolerant [54], so that failing nodes have the least impact on result data.

Situational Awareness approach, using a mechanism that has been borrowed from humans, can be applied in sensor data interpretation for Internet of Things (IoT), specifically, regarding processes of sensation, perception and cognition. In addition to specification-based and learning-based approaches, a perception-based approach utilizing Fuzzy Formal Concept was proposed [55] for Situational Awareness identification.

Semantic Sensor Network Ontology has been proposed in Reference [56] for managing interoperability between sensing systems. The Semantic Ground describes information for interoperability and cooperation among agents [57]. To enhance resilience in Semantic Sensor Networks, monitoring nodes may forward observations to association nodes, which develop Situational Awareness by mining association rules, for example, via a natural Artificial Bee Colony algorithm [57].

Electric Power Grids need efficient monitoring since, for outage detection, environmental monitoring, and fault diagnostics, different WSN-based approaches are reviewed [13]. Most of these approaches are also applicable in other kinds of applications.

5.3. Fault Isolation, Masking and Recovery

Subsequent to Fault Detection, Fault Identification, and Fault Diagnosis, a fault handling stage can be entered [49] to prevent further data corruption and system deterioration. The fault handling consists of Fault Isolation, Masking, and Recovery. Fault handling can hide the fault occurrence from other components by applying Fault Masking; the key techniques for such masking are informational, time, and physical redundancy [32]. Proposed masking technique For Underwater Vehicles is Triple Modular Redundancy (TMPR) [58], which is also one of the most commonly used Fault Masking techniques. Isolating a faulty component from the others can be facilitated by using virtualization [32]. In large scale distributed systems, frozen virtual images of healthy services have been used as checkpoints [59] for rolling back in case of a fault occurrence.

Fault Recovery ensures that the fault does not propagate to visible results, for instance, by rolling back to a previous healthy state (checkpointing) or re-trying failed operations (time redundancy). Some of the techniques for Fault Recovery can be Reconfiguration, which is changing the system's state so that the same or similar error is prevented from occurring again, and Adaptation, which is re-optimizing the system, for instance, after Reconfiguration task [33].

In Sensor Networks, different approaches for Fault Recovery have been used, that have different resource overheads, energy-efficiencies, scalabilities and network types. For both network and node Fault Recovery in wireless sensor networks, Mitra et al. (2016) [8] compares techniques, such as checkpoint-based recovery (CRAFT), agent-based recovery (ABSR), fault node recovery (FNR), cluster-based and hierarchical fault management (CHFM), and Failure Node Detection and Recovery algorithm (FNDRA). While some of those are specific to terrestrial wireless usage, some principles (e.g., checkpointing, etc.) can also be used in wired and/or underwater environments. To reduce the network bandwidth requirements, checkpoint backup can be mobile to nearby nodes [60] and used for recovering from fault situations.

In network protocols, Fault Masking and Fault Recovery are handled by error control schemes that are commonly categorized into the following three groups [2]:

- Automatic Repeat Request (ARQ)—re-transmission of corrupted data is asked;
- Forward Error Correction (FEC)—data corruption can be detected and corrected by the receiving end; and
- Hybrid ARQ (HARQ)—a combination of FEC and ARQ.

The cross-layer approach benefits Fault Recovery significantly since single-layer redundancy, such as hardware redundancy and application checkpointing, have very high costs, and latency between fault occurrence and detection makes the recovery difficult [33].

6. Comparative Analysis

All the papers that were selected according to the criteria described in Section 2 are listed in Table 2. The table includes information about the targeted extra-functional aspects and Fault Tolerance task(s). In addition, the Marine column in Table shows if the listed paper is explicitly touching aquatic environments. The papers are ordered by their order of citation within this survey paper. Papers that are not directly cited in the text but still listed in Table 2 are ordered chronologically by the publishing year. Papers that are not included in the analysis but are cited (e.g., definitions) have not been included in the table.

It can be seen from Table 2 that only two papers address both marine and cross-layer Fault Tolerance aspects. However, in the work targeting cross-layer analysis of error control [2], the term 'cross-layer' does not apply to the system stack but only to the communication protocol layers. Another work authored by the authors of this survey [61] is focusing on data-driven cross-layer Fault Tolerance. Thus, there is a serious gap in research addressing cross-layer Fault Tolerance in underwater sensor networks.

Regarding other extra-functional aspects, security in marine environments is addressed by six marine papers and is focusing on securing wireless communication [20,48,62], authentication [63], and hybrid attacks [64]. On scalability, seven marine papers were identified, and underwater scalability has been researched, for instance, in the context of monitoring underwater pipelines [22]. On Energy-efficiency, there were 14 Marine papers identified, and extensive focus has been on energy-efficient underwater wireless protocols [3,19,65–69] and less on other aspects. Open research issues from all the mentioned extra-functional aspects will be discussed in the following section.

Table 2. Categorized papers.

Pub.	1st Auth.	Year	Extra-Functional Aspect				Marine	Fault Tolerance (FT) Tasks and Other Research Areas
			Secure	Energy-Efficient	Scalable	Cross-Layer		
[1]	Domingo	2012	-	-	-	-	+	sensor network
[2]	Domingo	2012	-	-	-	+	+	sensor network, FT detect/recover, wireless
[3]	Zenia	2016	-	+	+	-	+	sensor network, routing protocol, survey, FT detect, FT recover
[4]	Jiang	2018	-	-	-	-	+	survey, wireless, sensor network
[5]	Kao	2017	-	-	-	-	+	FT design, survey, wireless
[6]	Veleski	2017	-	-	-	+	-	survey, FT detect, FT recover
[7]	Paradis	2007	-	-	-	+	-	FT detect/recover, survey, wireless
[8]	Mitra	2016	-	+	+	+	-	survey, wireless, FT detect, FT recover,
[9]	Atzori	2010	-	-	-	-	-	sensor network, survey
[10]	Diaz	2016	-	-	+	-	-	survey
[11]	Tyagi	2013	+	+	+	-	-	survey, wireless, routing protocol
[12]	Singh	2012	-	-	-	-	-	routing protocol, survey , wireless
[13]	Fadel	2015	-	-	-	-	-	survey, sensor network, wireless, FT detect
[14]	Moridi	2020	-	-	-	-	-	sensor network, wireless, FT detection, FT recovery
[15]	More	2017	-	+	+	-	-	sensor network, survey
[16]	Amin	2019	-	-	-	-	-	FT detect/recover, survey
[18]	Domingo	2009	-	-	-	-	+	FT detect, wireless
[19]	Xu	2012	-	+	+	-	+	FT detect, FT recover, sensor network
[20]	Lal	2016	+	-	+	-	+	wireless, sensor network
[21]	Das	2017	-	-	+	-	+	localization, sensor network, FT recover
[22]	Mohamed	2011	-	-	+	-	+	sensor network, FT detect
[23]	Kumar	2018	-	-	+	-	-	FT detect/recover
[25]	Khan	2013	-	-	+	+	-	FT detect/recover, wireless
[26]	Henkel	2011	-	-	-	+	-	FT design/detect/recover
[27]	Georgakos	2013	-	-	-	+	-	FT design/detect/recover, vehicle
[28]	Lorenz	2012	-	-	-	-	-	FT prevent
[29]	Sauli	2012	-	-	-	-	-	FT prevent
[30]	Rehman	2016	-	-	-	+	-	FT prevent/detect/recover
[31]	Kaaniche	2000	-	-	-	-	-	FT prevent/detect/recover
[33]	Carter	2010	-	-	-	+	-	FT design
[34]	Sofi	2018	-	+	-	-	+	sensor network, wireless
[35]	Noh	2016	-	-	-	-	+	routing protocol, sensor network, wireless
[37]	Wu	2007	-	-	-	-	-	deployment, localization, sensor network, wireless
[38]	Isler	2004	-	-	-	-	-	deployment, sensor network, wireless
[39]	Dong	2020	-	+	-	-	+	sensor network, wireless, FT recover

Table 2. *Cont.*

Pub.	1st Auth.	Year	Extra-Functional Aspect				Marine	Fault Tolerance (FT) Tasks and Other Research Areas
			Secure	Energy-Efficient	Scalable	Cross-Layer		
[40]	Cheng	2008	-	-	-	-	-	deployment, sensor network, wireless
[41]	Bhuvana	2018	-	+	-	-	-	sensor network, wireless, FT detect
[42]	Goyal	2017	-	-	-	-	+	wireless, sensor network
[43]	Goyal	2018	-	-	-	-	+	wireless, sensor network, FT detection, FT recovery
[45]	Campagnaro	2016	-	-	-	-	+	framework, wireless, sensor network
[46]	Petrioli	2015	-	-	-	-	+	framework wireless, sensor network
[47]	Petroccia	2013	-	-	-	-	+	framework, wireless, sensor network
[48]	Ateniese	2015	+	-	-	-	+	framework, wireless, sensor network
[49]	Liu	2013	+	+	-	+	-	wireless, sensor network
[51]	Gunawi	2011	-	-	+	-	-	FT design
[53]	Napolitano	1995	-	-	-	-	-	sensor network, FT detect, FT recover
[54]	Neti	1992	-	-	-	-	-	FT design
[55]	Benincasa	2014	-	-	-	-	-	sensor network
[56]	Compton	2012	-	-	-	-	-	sensor network, deployment
[57]	DAniello	2016	-	-	-	+	-	sensor network, FT detect, FT recover
[58]	Alansary	2019	-	-	-	-	+	vehicle, FT recovery
[59]	Cristea	2011	+	-	+	-	-	FT detect, FT recover
[60]	Salera	2007	-	-	-	+	-	sensor network, FT detect, FT recover
[61]	Vihman	2020	+	-	+	+	+	sensor network, FT detect
[62]	Han	2015	+	-	-	-	+	wireless, sensor network
[63]	Chae-Won	2016	+	-	-	-	+	sensor network, wireless
[64]	Han	2020	+	-	+	-	+	sensor network, wireless
[65]	Dong	2013	-	+	-	-	+	sensor network, wireless
[66]	Zhou	2016	-	+	-	-	+	, wireless, sensor network, routing protocol
[67]	Wang	2016	-	+	-	-	+	, sensor network, wireless
[68]	Huang	2011	-	+	-	-	+	wireless, sensor network, routing protocol
[69]	Rani	2017	-	+	-	-	+	sensor network, routing protocol
[70]	DeHon	2010	-	+	+	+	-	FT detect, FT recover
[71]	Darra	2017	+	-	-	-	-	survey, sensor network, wireless
[72]	Mitra	2010	-	-	-	+	-	FT detect, FT recover
[73]	Henkel	2014	-	-	-	+	-	FT detect, FT recover
[74]	Bulusu	2000	-	+	+	-	-	localization, sensor network
[75]	Nassif	2001	-	-	-	-	-	FT prevent
[76]	Zhao	2002	-	+	+	-	-	, wireless, sensor network
[77]	de Lemos	2004	-	-	-	-	-	FT design, sensor network
[78]	Bokareva	2005	-	-	-	+	-	cross-layer, FT design, FT recover, framework, sensor network
[79]	Heidemann	2006	-	-	-	-	+	sensor network, wireless
[80]	Mengjie	2007	-	-	+	+	-	wireless, sensor network, FT detect, FT recover
[81]	Lee	2008	-	-	-	-	-	wireless, FT detect, sensor network
[82]	Wang	2008	-	-	-	-	+	sensor network

Table 2. *Cont.*

Pub.	1st Auth.	Year	Extra-Functional Aspect				Marine	Fault Tolerance (FT) Tasks and Other Research Areas
			Secure	Energy-Efficient	Scalable	Cross-Layer		
[83]	Khan	2009	-	+	+	-	-	wireless, FT design, sensor network
[84]	Teymorian	2009	-	-	-	-	+	localization, sensor network
[85]	Yu	2009	-	-	-	-	+	localization, wireless, sensor network
[86]	Kim	2011	-	-	-	+	-	vehicle, FT detect, FT recover,
[87]	Tanasa	2011	-	-	-	-	-	vehicle, FT detect
[88]	Roman	2011	+	-	-	-	-	sensor network,
[89]	Paul	2011	+	-	-	-	-	sensor network
[90]	Xu	2011	-	-	-	-	+	wireless, sensor network, routing protocol, FT recovery
[91]	Thomas	2013	-	-	-	-	-	FT detect
[92]	Gubbi	2013	+	-	+	-	-	wireless, sensor network,
[93]	Guo	2013	-	-	+	-	+	localization, sensor network
[94]	Amory	2013	+	-	+	-	+	vehicle
[95]	Oteafy	2014	-	-	-	+	-	wireless, sensor network
[96]	Rault	2014	-	-	+	-	-	wireless, sensor network
[97]	Kuila	2014	-	+	-	-	-	wireless, sensor network, routing protocol
[98]	Zhu	2014	-	-	-	-	+	sensor network
[99]	Rossi	2015	-	+	-	-	+	sensor network, wireless
[100]	Bauer	2015	-	-	-	-	-	FT masking
[101]	Benson	2015	-	-	-	+	-	sensor network
[102]	Zhehao	2015	-	-	-	-	+	localization, wireless, sensor network
[103]	Han	2015	-	-	-	-	+	localization, wireless, sensor network, deployment
[104]	Valerio	2015	-	-	-	-	+	wireless, sensor network, routing protocol
[105]	Rehman	2016	-	-	-	+	-	FT detect, FT recover,
[106]	Sahoo	2016	-	-	-	+	-	FT design, FT detect
[107]	Li	2016	-	-	-	-	+	localization, vehicle
[108]	Liu	2016	-	-	-	-	+	sensor network, wireless, localization
[109]	Khan	2016	-	-	-	-	+	vehicle, sensor network
[110]	Koraz	2017	-	-	-	+	-	FT detect
[111]	Suvarna	2017	-	+	-	-	+	wireless, sensor network, routing protocol
[112]	Cario	2017	-	+	-	-	+	sensor network, wireless
[113]	Dong	2017	-	+	-	-	+	, localization, wireless, sensor network
[114]	Kao	2017	-	-	-	-	+	survey, sensor network, wireless
[115]	Mortazavi	2017	-	-	-	-	+	localization, wireless, sensor network
[116]	Seto	2017	-	-	-	-	+	vehicle
[117]	Azad	2018	-	-	-	+	-	FT detect, FT recovery
[118]	Sahu	2018	-	-	-	-	+	clustering, sensor network, routing protocol, FT detection, FT recovery
[119]	Dala	2018	-	-	-	-	+	sensor network, FT detection, FT recovery
[120]	Tang	2018	-	-	-	-	+	wireless, sensor network, fault, FT detection, FT recovery
[121]	Yanmaz	2018	-	-	+	-	-	vehicle, sensor network, wireless
[122]	Han	2018	-	-	-	-	+	localization, wireless, sensor network

Table 2. *Cont.*

Pub.	1st Auth.	Year	Extra-Functional Aspect Secure	Energy-Efficient	Scalable	Cross-Layer	Marine	Fault Tolerance (FT) Tasks and Other Research Areas
[123]	Shah	2018	-	-	+	-	+	localization, sensor network
[124]	Caporuscio	2020	-	-	-	-	-	sensor network, FT detection, FT recovery
[125]	Desai	2020	-	-	-	-	-	sensor network, FT detection
[126]	Jin	2020	-	-	-	-	+	sensor network, wireless, routing protocol, vehicle, FT detection; FT recovery
[127]	Prasanth	2020	-	+	-	-	+	wireless, sensor network, fault, ft recovery, ft detection

7. Open Research Issues

In the following, the open research issues identified are presented according to the categories of extra-functional aspects reported in Table 2.

7.1. Security

Faults and security are interrelated concepts [59]. It requires effort to prevent systems from being penetrated, even when they operate as intended; however, faults will add further uncertainty and make the task of prevention even harder. Faults can be created by an intrusion; but, moreover, faults can enable new intrusion vectors [70]—misbehaving devices violate key assumptions and create a number of new attack vectors to systems. For example, soft errors explained in Section 4.1 can be used to defeat cryptography [128]. In wireless sensor networks, intrusion detection systems have been investigated [71], and intrusion detection can be divided into Anomaly detection, which can work well for unknown attacks, and Misuse detection, for known attack signatures.

7.2. Energy-Efficiency

Power dissipation has by now reached a point where energy concerns limit the computation we can deploy on the chip [70], and the aim is shifting from transistor density and speed to energy density and cost. Energy density and efficiency need also to be addressed on a larger scale; for instance, WSNs may not have unlimited power supply and need to utilize energy-efficiency strategies [11,12,36,40]. For Fault Tolerance techniques, cross-layer approach is considered more energy-efficient [33] than single layer. Strategic redundancy in cross-layer approach may allow systems to safely operate on the verge of failure [70], spending less energy without going over the edge.

In sensor networks, energy consumption can be reduced, for instance, by using specific low-energy communication protocols, reducing the number and speed of the nodes, and pausing the nodes [129]. However, with the growing complexity of applications, energy consumption is becoming one of the limiting factors.

7.3. Scalability

One of the traditional benefits of scaling has been the decrease of cost per functionality [70], but easing reliability problems by multiplicating logic, voting and similar techniques means that the scaled technology might not offer a reduction of energy or area. Some Fault Tolerance techniques may increase computing overhead, and not all approaches are scalable [8]. Large scale fault tolerant systems are researched without paying special attention to energy and communication constraints [59].

7.4. Cross Layer Approach to Fault Tolerance

Faults are not going to disappear but likely to increase in the future [30]. One way to cope with faults is to accept imperfect devices to fail and compensate failures at higher levels in the system stack [70], tolerating faults across layers involving circuit design, firmware, operating system, applications, etc. Cross-layer fault tolerant systems have potential to implement reliable, high-performance and energy-efficient solutions without overwhelming costs [33] by distributing the responsibilities of tolerating faults across multiple layers [6]. Cross-layer Fault Tolerance has also been viewed from the perspective of sensor data layers [61].

In case Fault Detection and Fault Recovery are to be implemented in different system layers, then following challenges arise [72]:

- For statistical validation and metrics high confidence resource-light reliability and availability estimation is needed.
- Verification of resilience techniques, to be sure that resilience techniques perform under all possible scenarios.
- Reliability grades for testing and grading system-wide reliability and data integrity. Reliability may change under different workloads.

In addition to the cross-layer approach, a Multi-Layer approach [73] has also been proposed, where system layers are adapted to each other to reduce error propagation. However, in the opinion of the authors of the current paper, this does not constitute a principally distinct approach but, rather, an increment to the cross-layer approach.

8. Conclusions

The current paper presented a systematic survey on fault tolerant techniques in USNs and pointed out open research issues in this field. The paper considered fault tolerant techniques that are developed for underwater use or could be adapted for that. The techniques were divided into groups according to the taxonomy of Fault Tolerance tasks, and papers utilizing these techniques were discussed in sections corresponding to the tasks.

We collected top papers by conducting a systematic search from different online environments, related papers suggested by those environments, and sources cited by the collected papers. Next, we analyzed the collected papers, divided them into categories and discussed aspects covered in those papers. Areas of high research interest and open research issues in the scope of the initial criteria were detected and brought out. Additionally, in order to categorize and systematize the analyzed papers, taxonomies for fault sources and Fault Tolerance tasks were described, and a full table of the papers was presented.

The current paper is the first to investigate the state-of-the-art in Fault Tolerance, particularly cross-layer Fault Tolerance, in USNs. According to the survey, there is a lack of research covering the cross-layer Fault Tolerance aspect for underwater sensor networks. Therefore, the mentioned topic is a prospective candidate for future works on fault tolerant USNs.

Author Contributions: Conceptualization, L.V., M.K. and J.R.; methodology, L.V. and M.K.; software, L.V.; validation, L.V. and J.R.; formal analysis, L.V. ; investigation, L.V.; writing—original draft preparation, L.V., M.K. and J.R.; writing—review and editing, L.V and J.R.; visualization, L.V. and J.R.; supervision, M.K. and J.R.; funding acquisition, M.K. J.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Estonian Centre of Excellence in ICT Research (EXCITE) and by the Estonian Ministry of Education and Research and European Regional Fund (grant 2014-2020.4.01.20-0289).

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Domingo, M.C. An overview of the internet of underwater things. *J. Netw. Comput. Appl.* **2012**, *35*, 1879–1890, doi:10.1016/j.jnca.2012.07.012.
2. Domingo, M.C.; Vuran, M.C. Cross-layer analysis of error control in underwater wireless sensor networks. *Comput. Commun.* **2012**, *35*, 2162–2172, doi:10.1016/j.comcom.2012.07.010.
3. Zenia, N.Z.; Aseeri, M.; Ahmed, M.R.; Chowdhury, Z.I.; Shamim Kaiser, M. Energy-efficiency and reliability in MAC and routing protocols for underwater wireless sensor network: A survey. *J. Netw. Comput. Appl.* **2016**, *71*, 72–85, doi:10.1016/j.jnca.2016.06.005.
4. Jiang, S. State-of-the-Art Medium Access Control (MAC) Protocols for Underwater Acoustic Networks: A Survey Based on a MAC Reference Model. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 96–131, doi:10.1109/COMST.2017.2768802.
5. Kao, C.C.; Lin, Y.S.; Wu, G.D.; Huang, C.J. A study of applications, challenges, and channel models on the Internet of Underwater Things. In Proceedings of the 2017 International Conference on Applied System Innovation (ICASI), Sapporo, Japan, 13–17 May 2017; pp. 1375–1378, doi:10.1109/ICASI.2017.7988162.
6. Veleski, M.; Kraemer, R.; Krstic, M. An overview of cross-layer resilience design methods. In Proceedings of the Fifth International Conference Reliability—ETS'17 Fringe Workshop, Limasol, Cyprus, 25–26 May 2017.
7. Paradis, L.; Han, Q. A survey of fault management in wireless sensor networks. *J. Netw. Syst. Manag.* **2007**, *15*, 171–190, doi:10.1007/s10922-007-9062-0.
8. Mitra, S. Comparative study of fault recovery techniques in wireless sensor network. In Proceedings of the 2016 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Pune, India, 19–21 December 2016; pp. 19–21.
9. Atzori, L.; Iera, A.; Morabito, G. The Internet of Things: A survey. *Comput. Netw.* **2010**, *54*, 2787–2805, doi:10.1016/j.comnet.2010.05.010.
10. Díaz, M.; Martín, C.; Rubio, B. State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. *J. Netw. Comput. Appl.* **2016**, *67*, 99–117, doi:10.1016/j.jnca.2016.01.010.
11. Tyagi, S.; Kumar, N. A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks. *J. Netw. Comput. Appl.* **2013**, *36*, 623–645, doi:10.1016/j.jnca.2012.12.001.
12. Singh, S.P.; Sharma, S.C. A survey on cluster based routing protocols in wireless sensor networks. *Procedia Comput. Sci.* **2015**, *45*, 687–695, doi:10.1016/j.procs.2015.03.133.
13. Fadel, E.; Gungor, V.C.; Nassef, L.; Akkari, N.; Abbas Malik, M.G.; Almasri, S.; Akyildiz, I.F. A survey on wireless sensor networks for smart grid. *Comput. Commun.* **2015**, *71*, 22–33, doi:10.1016/j.comcom.2015.09.006.
14. Moridi, E.; Haghparast, M.; Hosseinzadeh, M.; Jassbi, S.J. Fault management frameworks in wireless sensor networks: A survey. *Comput. Commun.* **2020**, *155*, 205–226, doi:10.1016/j.comcom.2020.03.011.
15. More, A.; Raisinghani, V. A survey on energy efficient coverage protocols in wireless sensor networks. *J. King Saud Univ. Comput. Inf. Sci.* **2017**, *29*, 428–448, doi:10.1016/j.jksuci.2016.08.001.
16. Amin, A.A.; Hasan, K.M. A review of Fault Tolerant Control Systems: Advancements and applications. *Meas. J. Int. Meas. Confed.* **2019**, *143*, 58–68, doi:10.1016/j.measurement.2019.04.083.
17. Moher, D.; Liberati, A.; Tetzlaff, J.; Altman, D.G. Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *Int. J. Surg.* **2010**, *8*, 336–341, doi:10.1016/j.ijsu.2010.02.007.
18. Domingo, M.C. A topology reorganization scheme for reliable communication in underwater wireless sensor networks affected by shadow zones. *Sensors* **2009**, *9*, 8684–8708, doi:10.3390/s91108684.
19. Xu, J.; Li, K.; Min, G. Reliable and energy-efficient multipath communications in underwater sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 1326–1335, doi:10.1109/TPDS.2011.266.
20. Lal, C.; Petroccia, R.; Conti, M.; Alves, J. Secure underwater acoustic networks: Current and future research directions. In Proceedings of the 2016 IEEE Third Underwater Communications and Networking Conference (UComms), Lerici, Italy, 30 August–1 September 2016, doi:10.1109/UComms.2016.7583466.
21. Das, A.P.; Thampi, S.M. Fault-resilient localization for underwater sensor networks. *Ad Hoc Netw.* **2017**, *55*, 132–142, doi:10.1016/j.adhoc.2016.09.003.
22. Mohamed, N.; Jawhar, I.; Al-Jaroodi, J.; Zhang, L. Sensor network architectures for monitoring underwater pipelines. *Sensors* **2011**, *11*, 10738–10764, doi:10.3390/s111110738.
23. Kumar, S.; Balamurugan B. Fault tolerant cloud systems. In *Encyclopedia of Information Science and Technology*, 4th ed.; Khosrow-Pour, D.B.A., Ed.; IGI Global: Hershey, PA, USA 2018; Chapter 93, pp. 1075–1090, doi:10.4018/978-1-5225-2255-3.ch093.
24. Wilfredo, T.; Torres-Pomales, W. *Software Fault Tolerance: A Tutorial*; Technical Report October; NASA: Washington, DC, USA 2000.
25. Khan, M.Z. Fault Management in Wireless Sensor Networks. *GESJ Comput. Sci. Telecommun.* **2013**, *1, pp. 3–17*.
26. Henkel, J.; Hedrich, L.; Herkersdorf, A.; Kapitza, R.; Lohmann, D.; Marwedel, P.; Platzner, M.; Rosenstiel, W.; Schlichtmann, U.; Spinczyk, O.; et al. Design and architectures for dependable embedded systems. In Proceedings of the Seventh IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, Taipei, Taiwan, 9–14 October 2011; ACM Press: New York, NY, USA, 2011; p. 69, doi:10.1145/2039370.2039384.
27. Georgakos, G.; Schlichtmann, U.; Schneider, R.; Chakraborty, S. Reliability challenges for electric vehicles. In Proceedings of the 50th Annual Design Automation Conference, Austin, TX, USA, 29 May–7 June 2013; ACM Press: New York, NY, USA, 2013; p. 1, doi:10.1145/2463209.2488855.
28. Lorenz, D.; Barke, M.; Schlichtmann, U. Efficiently analyzing the impact of aging effects on large integrated circuits. *Microelectron. Reliab.* **2012**, *52*, 1546–1552, doi:10.1016/j.microrel.2011.12.029.

29. Sauli, Z.; Retnasamy, V.; Taniselass, S.; Shapri, A.H.; Hatta, R.M.; Aziz, M.H. Polymer core BGA vertical stress loading analysis. *Proc. Int. Conf. Comput. Intell. Model. Simul.* **2012**, *129*, 148–151, doi:10.1109/CIMSim.2012.83.
30. Rehman, S. Reliable Software for Unreliable Hardware—A Cross-Layer Approach. Ph.D. Thesis, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany, 2015.
31. Kaaniche, M.; Laprie, J.C.; Blanquart, J.P. Dependability engineering of complex computing systems. In Proceedings of the Sixth IEEE International Conference on Engineering of Complex Computer Systems 2000, Tokyo, Japan, 11–14 September 2000, pp. 36–46, doi:10.1109/iceccs.2000.873926.
32. Tanenbaum, A.S.; Van Steen, M. *Distributed Systems: Principles and Paradigms*; Prentice-Hall: Upper Saddle River, NJ, USA, 2007.
33. Carter, N.P.; Naeimi, H.; Gardner, D.S. Design techniques for cross-layer resilience. In Proceedings of the 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010), Dresden, Germany, 8–12 March 2010; pp. 1023–1028, doi:10.1109/DATE.2010.5456960.
34. Sofi, S.A.; Mir, R.N. Natural algorithm based adaptive architecture for underwater wireless sensor networks. In Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 22–24 March 2017; pp. 2343–2346, doi:10.1109/WiSPNET.2017.8300179.
35. Noh, Y.; Lee, U.; Lee, S.; Wang, P.; Vieira, L.F.; Cui, J.H.; Gerla, M.; Kim, K. HydroCast: Pressure routing for underwater sensor networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 333–347, doi:10.1109/TVT.2015.2395434.
36. Asim, M.; Mokhtar, H.; Merabti, M. A fault management architecture for wireless sensor network. In Proceedings of the 2008 International Wireless Communications and Mobile Computing Conference, Crete, Greece, 6–8 August 2008; pp. 779–785, doi:10.1109/IWCMC.2008.135.
37. Wu, C.H.; Lee, K.C.; Chung, Y.C. A Delaunay Triangulation based method for wireless sensor network deployment. *Comput. Commun.* **2007**, *30*, 2744–2752, doi:10.1016/j.comcom.2007.05.017.
38. Isler, V.; Kannan, S.; Daniilidis, K. Sampling based sensor-network deployment. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; pp. 1780–1785.
39. Dong, M.; Li, H.; Li, Y.; Deng, Y.; Yin, R. Fault-tolerant topology with lifetime optimization for underwater wireless sensor networks. *Sadhana Acad. Proc. Eng. Sci.* **2020**, *45*, doi:10.1007/s12046-020-01406-1.
40. Cheng, Z.; Perillo, M.; Heinzelman, W.B. General network lifetime and cost models for evaluating sensor network deployment strategies. *IEEE Trans. Mob. Comput.* **2008**, *7*, 484–497, doi:10.1109/TMC.2007.70784.
41. Bhuvana, V.P.; Preissl, C.; Tonello, A.M.; Huemer, M. Multi-Sensor Information Filtering with Information-Based Sensor Selection and Outlier Rejection. *IEEE Sens. J.* **2018**, *18*, 2442–2452, doi:10.1109/JSEN.2017.2789239.
42. Goyal, N.; Dave, M.; Verma, A.K. Data aggregation in underwater wireless sensor network: Recent approaches and issues. *J. King Saud Univ. Comput. Inf. Sci.* **2017**, doi:10.1016/j.jksuci.2017.04.007.
43. Goyal, N.; Dave, M.; Verma, A.K. A novel fault detection and recovery technique for cluster-based underwater wireless sensor networks. *Int. J. Commun. Syst.* **2018**, *31*, 1–14, doi:10.1002/dac.3485.
44. Kohnstamm, J.; Madhub, D. Mauritius Declaration. In Proceedings of the 36th International Conference of Data Protection & Privacy Commissioners, Balaclava, Mauritius, 14 October 2014; pp. 1–2.
45. Campagnaro, F.; Francescon, R.; Guerra, F.; Favaro, F.; Casari, P.; Diamant, R.; Zorzi, M. The DESERT underwater framework v2: Improved capabilities and extension tools. In Proceedings of the 2016 IEEE Third Underwater Communications and Networking Conference (UComms), Lerici, Italy, 30 August–1 September 2016, doi:10.1109/UComms.2016.7583420.
46. Petrioli, C.; Petroccia, R.; Potter, J.R.; Spaccini, D. The SUNSET framework for simulation, emulation and at-sea testing of underwater wireless sensor networks. *Ad Hoc Netw.* **2015**, *34*, 224–238, doi:10.1016/j.adhoc.2014.08.012.
47. Petroccia, R.; Spaccini, D. Comparing the SUNSET and DESERT frameworks for in field experiments in underwater acoustic networks. In *OCEANS 2013 MTS/IEEE Bergen: The Challenges of the Northern Dimension*; IEEE: Bergen, Norway, 2013, doi:10.1109/OCEANS-Bergen.2013.6608141.
48. Ateniese, G.; Capossele, A.; Gjanci, P.; Petrioli, C.; Spaccini, D. SecFUN: Security framework for underwater acoustic sensor networks. In *MTS/IEEE OCEANS 2015—Genova: Discovering Sustainable Ocean Energy for a New World*; IEEE: Genova, Italy, 2015, doi:10.1109/OCEANS-Genova.2015.7271735.
49. Liu, T.H.; Yi, S.C.; Wang, X.W. A fault management protocol for low-energy and efficient Wireless sensor networks. *J. Inf. Hiding Multimed. Signal Process.* **2013**, *4*, 34–45.
50. Case, J.D.; Fedor, M.; Schoffstall, M.L.; Davin, J. RFC1157: Simple Network Management Protocol (SNMP); Technical Report, RFC Editor. 1990.
51. Gunawi, H.S.; Do, T.; Hellerstein, J.M.; Stoica, I.; Borthakur, D.; Robbins, J. Failure as a service (faas): A cloud service for large-scale, online failure drills. In *Technical Report No. UCB/EECS-2011-87*; University of California: Berkeley, CA, USA, 2011.
52. Intel. Intelligent Platform Management Interface. 2004. Available online: https://www.intel.com/content/www/us/en/servers/ipmi/ipmi-home.html (accessed on 6 May 2021).
53. Napolitano, M.R.; Neppach, C.; Casdorph, V.; Naylor, S.; Innocenti, M.; Silvestri, G. Neural-network-based scheme for sensor failure detection, identification, and accommodation. *J. Guid. Control. Dyn.* **1995**, *18*, 1280–1286, doi:10.2514/3.21542.
54. Neti, C.; Schneider, M.H.; Young, E.D. Maximally Fault Tolerant Neural Networks. *IEEE Trans. Neural Netw.* **1992**, *3*, 14–23, doi:10.1109/72.105414.

55. Benincasa, G.; D'Aniello, G.; De Maio, C.; Loia, V.; Orciuoli, F. Towards perception-oriented situation awareness systems. *Adv. Intell. Syst. Comput.* **2014**, *322*, 813–824, doi:10.1007/978-3-319-11313-5_71.
56. Compton, M.; Barnaghi, P.; Bermudez, L.; García-Castro, R.; Corcho, O.; Cox, S.; Graybeal, J.; Hauswirth, M.; Henson, C.; Herzog, A.; et al. The SSN ontology of the W3C semantic sensor network incubator group. *J. Web Semant.* **2012**, *17*, 25–32, doi:10.1016/j.websem.2012.05.003.
57. D'Aniello, G.; Gaeta, A.; Orciuoli, F. Artificial bees for improving resilience in a sensor middleware for Situational Awareness. In Proceedings of the 2015 Conference on Technologies and Applications of Artificial Intelligence, Tainan, Taiwan, 20–22 November 2015; pp. 300–307, doi:10.1109/TAAI.2015.7407104.
58. Alansary, K.A.; Daoud, R.M.; Amer, H.H.; Elsayed, H.M. Networked control system architecture for autonomous underwater vehicles with redundant sensors. In Proceedings of the 2019 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 27–29 June 2019; pp. 2019–2022, doi:10.1109/ECAI46879.2019.9042033.
59. Cristea, V.; Dobre, C.; Pop, F.; Stratan, C.; Costan, A.; Leordeanu, C.; Tirsa, E. A dependability layer for large-scale distributed systems. *Int. J. Grid Util. Comput.* **2011**, *2*, 109, doi:10.1504/IJGUC.2011.040598.
60. Salera, I.; Agbaria, A.; Eltoweissy, M. Fault-tolerant mobile sink in networked sensor systems. In Proceedings of the 2006 2nd IEEE Workshop on Wireless Mesh Networks, Reston, VA, USA, 25–28 September 2007; pp. 106–108, doi:10.1109/WIMESH.2006.288624.
61. Vihman, L.; Kruusmaa, M.; Raik, J. Data-driven cross-layer fault management architecture for sensor networks. In Proceedings of the 2020 16th European Dependable Computing Conference (EDCC), Munich, Germany, 7–10 September 2020; pp. 33–40, doi:10.1109/EDCC51268.2020.00015.
62. Han, G.; Jiang, J.; Sun, N.; Shu, L. Secure communication for underwater acoustic sensor networks. *IEEE Commun. Mag.* **2015**, *53*, 54–60, doi:10.1109/MCOM.2015.7180508.
63. Chae-Won, Y.; Jae-Hoon, L.; Okyeon, Y.; Soo-Hyun, P. Ticket-based authentication protocol for Underwater Wireless Sensor Network. In Proceedings of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 215–217, doi:10.1109/ICUFN.2016.7537019.
64. Han, G.; He, Y.; Jiang, J.; Wang, H.; Peng, Y.; Fan, K. Fault-Tolerant Trust Model for Hybrid Attack Mode in Underwater Acoustic Sensor Networks. *IEEE Netw.* **2020**, *34*, 330–336, doi:10.1109/MNET.001.2000006.
65. Dong, Y.; Dong, H. Simulation study on cross-layer design for energy conservation in underwater acoustic networks. In *2013 OCEANS—San Diego*; IEEE: San Diego, CA, USA, 2013; pp. 1–4, do:10.23919/OCEANS.2013.6741315
66. Zhou, Z.; Yao, B.; Xing, R.; Shu, L.; Bu, S. E-CARP: An Energy Efficient Routing Protocol for UWSNs in the Internet of Underwater Things. *IEEE Sens. J.* **2016**, *16*, 4072–4082, doi:10.1109/JSEN.2015.2437904.
67. Wang, K.; Gao, H.; Xu, X.; Jiang, J.; Yue, D. An Energy-Efficient Reliable Data Transmission Scheme for Complex Environmental Monitoring in Underwater Acoustic Sensor Networks. *IEEE Sens. J.* **2016**, *16*, 4051–4062, doi:10.1109/JSEN.2015.2428712.
68. Huang, C.J.; Wang, Y.W.; Liao, H.H.; Lin, C.F.; Hu, K.W.; Chang, T.Y. A power-efficient routing protocol for underwater wireless sensor networks. *Appl. Soft Comput. J.* **2011**, *11*, 2348–2355, doi:10.1016/j.asoc.2010.08.014.
69. Rani, S.; Hassan, S.; Malhotra, J.; Talwar, R. Energy e ffi cient chain based routing protocol for underwater wireless sensor networks. *J. Netw. Comput. Appl.* **2017**, *92*, 1–9, doi:10.1016/j.jnca.2017.01.011.
70. DeHon, A.; Quinn, H.M.; Carter, N.P. Vision for cross-layer optimization to address the dual challenges of energy and reliability. In Proceedings of the 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010), Dresden, Germany, 8–12 March 2010; pp. 1017–1022, doi:10.1109/DATE.2010.5456959.
71. Darra, E.; Katsikas, S.K. A survey of intrusion detection systems in wireless sensor networks. *Intrusion Detect. Prev. Mob. Ecosyst.* **2017**, 393–458, doi:10.1201/b21885.
72. Mitra, S.; Brelsford, K.; Sanda, P.N. Cross-layer resilience challenges: Metrics and optimization. In Proceedings of the 2010 Design, Automation & Test in Europe Conference & Exhibition (DATE 2010), Dresden, Germany, 8–12 March 2010; pp. 1029–1034, doi:10.1109/DATE.2010.5456961.
73. Henkel, J.; Bauer, L.; Zhang, H.; Rehman, S.; Shafique, M. Multi-layer dependability: From microarchitecture to application level. In Proceedings of the 51st Annual Design Automation Conference, San Francisco, CA, USA, 1–5 June 2014; pp. 47:1–47:6, doi:10.1145/2593069.2596683.
74. Bulusu, N.; Heidemann, J.; Estrin, D. GPS-less low-cost outdoor localization for very small devices. *IEEE Pers. Commun.* **2000**, *7*, 28–34, doi:10.1109/98.878533.
75. Nassif, S. Modeling and analysis of manufacturing variations. In Proceedings of the IEEE 2001 Custom Integrated Circuits Conference, San Diego, CA, USA, 9 May 2001; pp. 223–228, doi:10.1109/CICC.2001.929760.
76. Zhao, Y.J.; Govindan, R.; Estrin, D. Residual energy scan for monitoring sensor networks. In Proceedings of the 2002 IEEE Wireless Communications and Networking Conference Record, Orlando, FL, USA, 17–21 March 2002; Volume 1, pp. 356–362, doi:10.1109/WCNC.2002.993521.
77. de Lemos, R.; Gacek, C.; Romanovsky, A. LNCS 3069—Architecting dependable systems II. In *Architecting Dependable Systems II*; Springer-Verlag: Berlin, Germany; 2004; pp. 286–304.
78. Bokareva, T.; Bulusu, N.; Jha, S. SASHA: Toward a self-healing hybrid sensor network architecture. In Proceedings of the Second IEEE Workshop on Embedded Networked Sensors 2005, Sydney, Australia, 31 May 2005; Volume 2005, pp. 71–78, doi:10.1109/EMNETS.2005.1469101.

79. Heidemann, J.; Wei, Y.; Wills, J.; Syed, A.; Yuan, L. Research challenges and applications for underwater sensor networking. In Proceedings of the IEEE Wireless Communications and Networking Conference 2006, Las Vegas, NV, USA, 3–6 April 2006; pp. 228–235, doi:10.1109/WCNC.2006.1683469.
80. Mengjie, Y.; Mokhtar, H.; Merabti, M. Fault management in wireless sensor networks. *IEEE Wirel. Commun.* **2007**, *14*, 13–19, doi:10.1111/1471-0528.14614.
81. Lee, M.H.; Choi, Y.H. Fault detection of wireless sensor networks. *Comput. Commun.* **2008**, *31*, 3469–3475, doi:10.1016/j.comcom.2008.06.014.
82. Wang, Y.; Cao, L.; Dahlberg, T.A. Efficient fault tolerant topology control for three-dimensional wireless networks. In Proceedings of the 2008 17th International Conference on Computer Communications and Networks, St. Thomas, VI, USA, 3–7 August 2008; pp. 336–341, doi:10.1109/ICCCN.2008.ECP.75.
83. Khan, M.Z.; Merabti, M.; Askwith, B. Design considerations for fault management in wireless sensor networks. In Proceedings of the 10th Annual Conference on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool, UK, 12–23 June 2009, doi:10.13140/2.1.1071.9685.
84. Teymorian, A.Y.; Member, S.; Cheng, W.; Ma, L.; Cheng, X. 3D Underwater Sensor Network Localization. *Network* **2009**, *8*, 1610–1621, doi:10.1109/TMC.2009.80.
85. Yu, C.H.; Lee, K.H.; Moon, H.P.; Choi, J.W.; Seo, Y.B. Sensor localization algorithms in underwater wireless sensor networks. In Proceedings of the 2009 ICCAS-SICE, Fukuoka, Japan, 18–21 August 2009; pp. 1760–1764.
86. Kim, J.Y.; Yoon, H.; Kim, S.H.; Son, S.H. Fault management of robot software components based on OPRoS. In Proceedings of the 2011 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, Newport Beach, CA, USA, 28–31 March 2011; pp. 253–260, doi:10.1109/ISORC.2011.41.
87. Tanasa, B.; Bordoloi, U.D.; Eles, P.; Peng, Z. Reliability-aware frame packing for the static segment of flexray. In Proceedings of the 2011 Proceedings of the Ninth ACM International Conference on Embedded Software (EMSOFT), Taipei, Taiwan, 9–14 October 2011; ACM Press: New York, NY, USA, 2011; pp. 175–184, doi:10.1145/2038642.2038670.
88. Roman, R.; Najera, P.; Lopez, J. Securing the Internet of things. *Comput. (Long. Beach. Calif.)* **2011**, *44*, 51–58, doi:10.1109/MC.2011.291.
89. Paul, S.; Pan, J.; Jain, R. Architectures for the future networks and the next generation Internet: A survey. *Comput. Commun.* **2011**, *34*, 2–42, doi:10.1016/j.comcom.2010.08.001.
90. Xu, M.; Liu, G. Fault tolerant routing in three-dimensional underwater acoustic sensor networks. In Proceedings of the 2011 International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 9–11 November 2011; doi:10.1109/WCSP.2011.6096859.
91. Thomas, A.; Pattabiraman, K. Error detector placement for soft computation. *Proc. Int. Conf. Dependable Syst. Netw.* **2013**, *15*, 1–25, doi:10.1109/DSN.2013.6575353.
92. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M. Internet of Things (IoT): A vision, architectural elements, and future directions. *Futur. Gener. Comput. Syst.* **2013**, *29*, 1645–1660, doi:10.1016/j.future.2013.01.010.
93. Guo, Y.; Liu, Y. Localization for anchor-free underwater sensor networks. *Comput. Electr. Eng.* **2013**, *39*, 1812–1821, doi:10.1016/j.compeleceng.2013.02.001.
94. Amory, A.; Meyer, B.; Osterloh, C.; Tosik, T.; Maehle, E. Towards fault-tolerant and energy-efficient swarms of underwater robots. In Proceedings of the 2013 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum, Cambridge, MA, USA, 20–24 May 2013; pp. 1550–1553, doi:10.1109/IPDPSW.2013.215.
95. Oteafy, S.M.; Hassanein, H.S. Resilient Sensor Networks utilizing dynamic resource reuse: A novel paradigm towards fault-tolerant wireless sensing. In Proceedings of the 2014 International Conference on Computing, Management and Telecommunications (ComManTel), Da Nang, Vietnam, 27–29 April 2014; pp. 217–222, doi:10.1109/ComManTel.2014.6825607.
96. Rault, T.; Bouabdallah, A.; Challal, Y. Energy efficiency in wireless sensor networks: A top-down survey. *Comput. Netw.* **2014**, *67*, 104–122, doi:10.1016/j.comnet.2014.03.027.
97. Kuila, P.; Jana, P.K. Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach. *Eng. Appl. Artif. Intell.* **2014**, *33*, 127–140, doi:10.1016/j.engappai.2014.04.009.
98. Zhu, Y.; Lu, X.; Pu, L.; Su, Y.; Martin, R.; Zuba, M.; Peng, Z.; Cui, J.H. Aqua-Sim: An NS-2 Based Simulator for Underwater Sensor Networks. In *OCEANS 2009*; IEEE: Biloxi, MS, USA , 2014; p. 2014, doi:10.23919/OCEANS.2009.5422081.
99. Rossi, P.S.; Member, S.; Ciuonzo, D. Energy Detection for MIMO Decision Fusion in Underwater Sensor Networks. *IEEE Sens. J.* **2015**, *15*, 1630–1640.
100. Bauer, L.; Henkel, J.; Herkersdorf, A.; Kochte, M.A.; Kühn, J.M.; Rosenstiel, W.; Schweizer, T.; Wallentowitz, S.; Wenzel, V.; Wild, T.; et al. Adaptive multi-layer techniques for increased system dependability. *It-Inf. Technol.* **2015**, *57*, 149–158, doi:10.1515/itit-2014-1082.
101. Benson, K. Enabling resilience in the Internet of Things. In Proceedings of the 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), St. Louis, MO, USA, 23–27 March 2015; pp. 230–232, doi:10.1109/PERCOMW.2015.7134032.
102. Zhehao, W.; Xia, L. An Improved Underwater Acoustic Network Localization Algorithm. *China Commun.* **2015**, *12*, 77 – 83.
103. Han, G.; Zhang, C.; Shu, L.; Rodrigues, J.J.P.C. Impacts of Deployment Strategies on Localization Performance in Underwater Acoustic Sensor Networks. *IEEE Trans. Ind. Electron.* **2015**, *62*, 1725–1733, doi:10.1109/TIE.2014.2362731.

104. Valerio, V.D.; Petrioli, C.; Pescosolido, L.; Van Der Shaar, M. A reinforcement learning-based data-link protocol for underwater acoustic communications. In Proceedings of the 10th International Conference on Underwater Networks & Systems, Arlington, VA, USA, 22–24 October 2015; pp. 1–5, doi:10.1145/2831296.2831338.
105. Rehman, S.; Chen, K.H.; Kriebel, F.; Toma, A.; Shafique, M.; Chen, J.J.; Henkel, J. Cross-Layer Software Dependability on Unreliable Hardware. *IEEE Trans. Comput.* **2016**, *65*, 80–94, doi:10.1109/TC.2015.2417554.
106. Sahoo, S.S.; Veeravalli, B.; Kumar, A. Cross-layer fault-tolerant design of real-time systems. In Proceedings of the 2016 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Storrs, CT, USA, 19–20 September 2016; pp. 63–68, doi:10.1109/DFT.2016.7684071.
107. Li, J.; Gao, H.; Zhang, S.; Chang, S.; Chen, J.; Liu, Z. Self-localization of autonomous underwater vehicles with accurate sound travel time solution. *Comput. Electr. Eng.* **2016**, *50*, 26–38, doi:10.1016/j.compeleceng.2015.11.018.
108. Liu, J.; Wang, Z.; Cui, J.H.; Zhou, S.; Yang, B. A Joint Time Synchronization and Localization Design for Mobile Underwater Sensor Networks. *IEEE Trans. Mob. Comput.* **2016**, *15*, 530–543, doi:10.1109/TMC.2015.2410777.
109. Khan, J.U.; Cho, H.S. A multihop data-gathering scheme using multiple AUVs in hierarchical underwater sensor networks. *Int. Conf. Inf. Netw.* **2016**, 265–267, doi:10.1109/ICOIN.2016.7427074.
110. Koraz, Y.; Gabbar, H.A. Bayesian-neuro-fuzzy network based online condition monitoring system for resilient micro energy grid using FPGA. In Proceedings of the 2017 IEEE 7th International Conference on Power and Energy Systems (ICPES), Toronto, ON, Canada, 1–3 November 2017; pp. 85–89, doi:10.1109/ICPESYS.2017.8215926.
111. Suvarna, M.; Patil, A.; Mishra, M.P. Improved Mobicast Routing Protocol to Minimize Energy Consumption for Underwater Sensor Networks. *Int. J. Res. Sci. Eng.* **2017**, *3*, 201.
112. Cario, G.; Casavola, A.; Gjanci, P.; Lupia, M.; Petrioli, C.; Spaccini, D. Long lasting underwater wireless sensors network for water quality monitoring in fish farms. In *OCEANS 2017—Aberdeen*, IEEE: Aberdeen, UK; 2017; pp. 1–6, doi:10.1109/OCEANSE.2017.8084777.
113. Dong, Y.; Wang, R.; Li, Z.; Cheng, C.; Zhang, K. Poster abstract: Improved reverse localization schemes for underwater wireless sensor networks. In Proceedings of the 2017 16th ACM/IEEE International Conference on Information Processing in Sensor Networks, Pittsburgh, PA, USA, 18–21 April 2017; Volume 1, pp. 323–324, doi:10.1145/3055031.3055070.
114. Kao, C.C.; Lin, Y.S.; Wu, G.D.; Huang, C.J. A Comprehensive Study on the Internet of Underwater Things: Applications, Challenges, and Channel Models. *Sensors* **2017**, *17*, 1477, doi:10.3390/s17071477.
115. Mortazavi, E.; Javidan, R.; Dehghani, M.J.; Kavoosi, V. A robust method for underwater wireless sensor joint localization and synchronization. *Ocean Eng.* **2017**, *137*, 276–286, doi:10.1016/j.oceaneng.2017.04.006.
116. Seto, M.L.; Bashir, A.Z. Fault tolerance considerations for long endurance AUVs. *Proc. Annu. Reliab. Maintainab. Symp.* **2017**, doi:10.1109/RAM.2017.7889661.
117. Azad, S.P. Cross-Layer Dependability Management in Network on Chip Based System on Chip. Ph.D. Thesis, Tallinn Technical University, Tallinn, Estonia, 13 March 2018.
118. Sahu, B.; Sahu, P.; Dash, P. Fault tolerant dynamic multi-hop clustering in under water sensor network. In Proceedings of the 2018 International Conference on Information Technology (ICIT), Bhubaneswar, India, 19–21 December 2018; pp. 101–105, doi:10.1109/ICIT.2018.00031.
119. Dala, A.; Adetomi, A.; Enemali, G.; Arslan, T. RR4DSN: Reconfigurable Receiver for Deepwater Sensor Nodes. In Proceedings of the 2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS), Edinburgh, UK, 6–9 August 2018; pp. 280–284, doi:10.1109/AHS.2018.8541455.
120. Tang, Y.; Chen, Y.; Yu, W.; Xu, X. Emergency communication schemes for muliti-hop underwater acoustic cooperartive networks. In Proceedings of the 2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Qingdao, China, 14–16 September 2018; doi:10.1109/ICSPCC.2018.8567788.
121. Yanmaz, E.; Yahyanejad, S.; Rinner, B.; Hellwagner, H.; Bettstetter, C. Drone networks: Communications, coordination, and sensing. *Ad Hoc Netw.* **2018**, *68*, 1–15, doi:10.1016/j.adhoc.2017.09.001.
122. Han, Y.; Zhang, J.; Sun, D. Error control and adjustment method for underwater wireless sensor network localization. *Appl. Acoust.* **2018**, *130*, 293–299, doi:10.1016/j.apacoust.2017.08.007.
123. Shah, M.; Wadud, Z.; Sher, A.; Ashraf, M.; Khan, Z.A.; Javaid, N. Position adjustment-based location error-resilient geo-opportunistic routing for void hole avoidance in underwater sensor networks. *Concurr. Comput.* **2018**, 1–18, doi:10.1002/cpe.4772.
124. Caporuscio, M.; Flammini, F.; Khakpour, N.; Singh, P.; Thornadtsson, J. Smart-troubleshooting connected devices: Concept, challenges and opportunities. *Futur. Gener. Comput. Syst.* **2020**, *111*, 681–697, doi:10.1016/j.future.2019.09.004.
125. Desai, N.; Punnekkat, S. Enhancing fault detection in time sensitive networks using machine learning. In Proceedings of the 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), Bengaluru, India, 7–11 January 2020; pp. 714–719, doi:10.1109/COMSNETS48256.2020.9027357.
126. Jin, Z.; Zhao, Q.; Luo, Y. Routing Void Prediction and Repairing in AUV-Assisted Underwater Acoustic Sensor Networks. *IEEE Access* **2020**, *8*, 54200–54212, doi:10.1109/ACCESS.2020.2980043.
127. Prasanth, A. Certain Investigations on Energy-Efficient Fault Detection and Recovery Management in Underwater Wireless Sensor Networks. *J. Circuits Syst. Comput.* **2020**, doi:10.1142/S0218126621501371.

128. Xu, J.; Chen, S.; Kalbarczyk, Z.; Iyer, R.K. An experimental study of security vulnerabilities caused by errors. *Proc. Int. Conf. Dependable Syst. Netw.* **2001**, 421–430, doi:10.1109/DSN.2001.941426.
129. Safara, F.; Souri, A.; Baker, T.; Al Ridhawi, I.; Aloqaily, M. PriNergy: A priority-based energy-efficient routing method for IoT systems. *J. Supercomput.* **2020**, *76*, 8609–8626, doi:10.1007/s11227-020-03147-8.

Appendix 3

III

L. Vihman, T. M. Parts, H. K. Aljas, M. Thalfeldt, and J. Raik. Algorithms for online CO2 baseline correction in intermittently occupied rooms. In *18th Heal. Build. Eur. Conf.*, Aachen, Germany, 2023. Emerald

Algorithms for online CO_2 baseline correction in intermittently occupied rooms

Lauri Vihman [a,b], *Tuule Mall* Parts [a,c], *Hans Kristjan* Aljas [a,c], *Martin* Thalfeldt [a,c], *Jaan* Raik [a,b]

[a] FinEst Centre for Smart Cities (Finest Centre), Tallinn University of Technology, Tallinn, Estonia, tuule.parts@taltech.ee.

[b] Centre for Dependable Computing Systems, Department of Computing Systems, Tallinn University of Technology, Estonia., lauri.vihman@taltech.ee.

[c] Department of Civil Engineering and Architecture, Tallinn University of Technology, Tallinn, Estonia.

Abstract. CO_2 sensor data is often applied for Demand Controlled Ventilation (DCV), Indoor Air Quality (IAQ) assessment, and occupancy detection. In room controllers, the auto-calibration function shifts the zero level so that the measurements would not drift off. However, this creates jumps in data and sometimes values below outdoor CO_2 level. If the data is further used, occupancy is detected differently, the ventilation would not function as designed, and the assessment would result in different IAQ class certificates. Therefore, in this work, a statistical method to correct the CO_2 baseline automatically in real time was developed based on measurements from a school building in Estonia in 56 different rooms. The school had balanced heat recovery ventilation that assured adequate ventilation. During the process, the performance of different algorithms and parameters for the correction were compared. The CO_2 concentration baseline correction algorithm was realised using the 1% percentile and 10-hour sliding time window as an optimal compromise to correct the base level to 400 ppm and the algorithm performed well based on qualitative assessment. The impact of the algorithm was significant when comparing the initially logged and corrected values against CO_2 concentration thresholds 550, 800, and 1000 ppm.

Keywords. Sensor auto-calibration, indoor air quality, data quality, CO_2 sensors, CO_2 monitoring

1. Introduction

Latest proposal for revising Energy Performance of Buildings Directive (EPBD) (European Commission, 2021) states that Member States shall require zero-emission buildings to be equipped with measuring and control devices for the monitoring and regulation of Indoor Air Quality (IAQ). In existing buildings, installation of such devices shall be required, where technically and economically feasible, when a building undergoes a major renovation. There has been discussion about which building types to include and if monitoring requirement is justified, but nevertheless carbon dioxide (CO_2) sensors are needed for regulation of IAQ and they are installed in an increasing number of new and deeply renovated non-residential buildings.

Measuring concentration of gases is technically complicated and in practice the quality of raw data from CO_2 sensors should be verified and post-processed if needed for further analysis. Mylonas et al. (Mylonas, Kazanci, Andersen, & Olesen, 2019) measured accuracy of temperature, relative humidity and CO_2 sensors from 6 producers in laboratory conditions. The measurement errors of CO_2 sensors ranged between -413 and +4589 ppm and in case of some sensor the measurement error strongly correlated with the air temperature. The authors concluded that significant improvement on CO_2 measurement capabilities are required, before these types of sensors can be installed in buildings for CO_2 concentration control. Nevertheless, CO_2 sensors are increasingly often used in buildings and it is currently difficult to assure the quality of sensors. This can often lead to inadequate IAQ regulation and assessment.

The aim of this study is to develop a data-driven method for post-processing of raw data from CO_2 sensors to improve the quality of ventilation control and IAQ assessment. Data collected by 56 CO_2 sensors from a school building with a balanced heat recovery ventilation in Estonia was analysed and methods for correcting the base level of CO_2 concentration and data cleaning were developed. Additionally, the temperature dependency of measured CO_2 concentrations from room temperatures were analysed. The developed methods can be implemented in existing building management and IAQ analysis tools with reasonable computational cost.

The study was overall conducted in the following steps:

- Data collection
- Initial expert assessment of data and classification of typical patterns
- CO_2 baseline correction method development:
    - Initial comparison of methods for baseline correction
    - In-depth analysis of promising methods and parameter tuning
- Expert assessment of selected method for baseline correction results

Sections 2 describes the case study building, data collection and expert assessment principles of raw and processed data. The CO_2 data correction method development is described in Section 3.

2. Data collection, occupancy detection and expert assessment

2.1 Building description and data collection

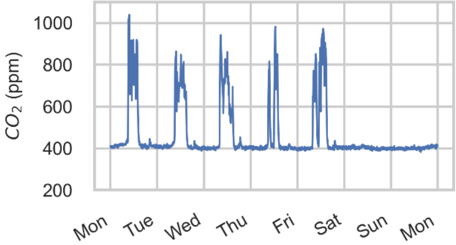
The data was collected from 56 rooms of a high school building from September 1st, 2020 until March 3rd, 2021. Air temperature, relative humidity, and CO_2 level was measured every 5 minutes. The measuring period includes both normal operation and periods when the building was not used, so different patterns can be observed. The school building had a balanced heat recovery ventilation system installed that was also used during the measurement period.

Commercial indoor air quality sensors SMT-IAQ3 ("Indoor Air Quality Monitor," 2020) were used. The producer claims temperature sensor accuracy to be +/- 0.5°C at 25°C (operating range –5 to 50°C), and relative humidity (RH) sensor accuracy to be +/- 5% at 25°C and 30% to 80% RH (range 0 to 95 % RH). The CO_2 sensor is a non-dispersive infrared sensor with +/- 30ppm accuracy at 25°C, and the operating range is 0-2000 ppm. The sensor applies auto-calibration as the manual ("Indoor Air Quality Monitor," 2020) states that – "*The CO_2 sensor within the SMT-IAQ3 has an advanced learning self-calibrating function. This calibration process takes place over an 8-day period.*"

2.2 Expert assessment

The initial expert assessment of collected data was conducted to identify the occurrence of outliers and unexpected patterns. The logged CO_2 concentration time series were plotted and qualitatively assessed. The expected behaviour of CO_2 concentration in rooms was values fluctuating around 800±300 ppm during expected occupancy hours of 8:00-16:00 on workdays and values fluctuating around 400±50 ppm outside occupancy hours (see **Figure *1***).

Figure 1. *Expected behaviour of CO_2 concentration*



The following typical cases of deviation from the expected behaviour were identified and were subject for further analysis and algorithm development:

- ***Outliers*** were identified as sudden change of the logged CO_2 concentration, which was followed by a sudden change of the same magnitude in the opposite direction.
    - ***Incorrect baseline*** was identified as a significant difference of CO_2 concentration from the outdoor concentration of 400 ppm.
    - ***Auto-calibration*** of the indoor climate sensor controller was identified as a sudden change in the logged CO_2 concentration to approximately 400 ppm with no change in the opposite direction afterwards.
- ***Potentially inadequate placement of CO_2 sensor*** was identified as sudden changes in the CO_2 concentration at reoccurring times of workdays when the ventilation system is either turned on or off. This could be caused by non-uniform distribution of CO_2 in the room air caused by the air distribution solution.
- ***CO_2 concentration's dependency on the air temperature*** was identified as unexplainable CO_2 concentration fluctuation during unoccupied hours that had negative correlation with the air temperature fluctuation.

The typical examples of expected behaviour are illustrated in Section 4.1.

2.3 Occupancy detection

The expert assessment was carried for both unoccupied and occupied periods of classrooms and for that an automated process for occupancy detection was implemented. The occupancy detection algorithm is an adaptation of Pedersen et al. method, which is based on CO_2 concentration trajectory in rooms (Pedersen, Nielsen, & Petersen, 2017). However, due to a lower sampling frequency in the current case study and performance related implementation difficulties, an alternative method was developed based on the existing one.

The method uses a combination of bilateral filtering (Thompson, 2014) and dynamic gradient-based signal trajectory classification. Bilateral filter preserves the transients, while removing unwanted sensor noise. The filtered signal is then classified into three categories: rising, stalling and decreasing, based on the gradient. Rising signal or rising stalling signal after rising signal represent occupancy. However, the gradient thresholds for this classification change dynamically between manually calibrated boundaries as a piecewise linear function of absolute CO_2 level at a given timestep. The calibration of these thresholds depends also on data timestep size, since the gradients in data with lower resolution also tend to be lower. When the level is near the baseline, greater rise and smaller decrease gradients are expected to trigger their respective states. At CO_2 levels above 700ppm, smaller rise and greater decrease are required. This allows to preserve more details in the bilateral filtering stage, without triggering false positives due to remaining noise. Less filtered signal, smaller rise and greater decrease thresholds are important to avoid false negatives during elevated CO_2 levels. Since the method uses absolute CO_2 values for this task, it is important that the CO_2 baseline correction is always performed before occupancy detection.

3. Method development

3.1 Initial comparison of methods

Although it would be better to correct CO_2 baseline in the source and having correct data through all the data lifecycle i.e. inside the sensor where it is measured, many of the used sensors are commercial with precisely unknown compensation mechanisms and faults and accessing or updating their internals is impossible. Thus, we are offering to fix the data at later processing stages while still retaining the possibility of doing it in real-time.

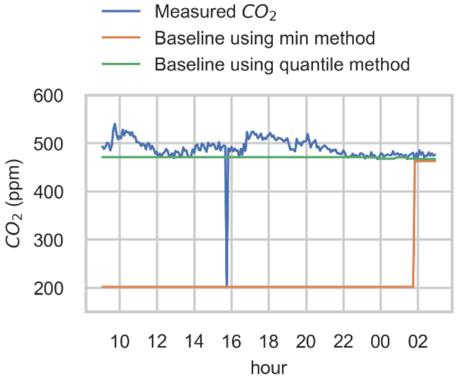
CO_2 value cannot be significantly lower than outdoors CO_2. When room is occupied the CO_2 level can rise but should return gradually to outdoor base level when unoccupied due to ventilation systems and natural air exchange. Thus, taking above consideration into account we tried to use the minimal value of a sliding time window to correct shifting baseline. However, there could be some outlier measurements occurring that reach unrealistically low values which make baseline correction using this method erroneous.

CO_2 measurements from a school classroom is shown on **Figure 2** and additionally, baselines calculated using 10 hour sliding window timeframe. It can be seen that one outlier measurement reaches approximately 200 ppm which is unrealistic value and calculated baseline using minimal value of sliding time window moving also to unrealistic value. These low peak outliers occur seldom in our data, but they exist. In case of occurring, there is usually only one outlier value, neighbouring values are assumed to be correct.

To overcome this problem, data pruning or smoothing methods can be used to remove the outlier data before using minimal value. There is also possibility of using a low quantile method (Hyndman & Fan, 1996). The reasoning behind using quantile is that most rooms are unoccupied some of the time (e.g. at night) returning to base value. If none or very few of the measurements in a certain time interval have been less than a certain value, then this can be considered as current base value.

It is possible to distinguish online (sequential) and offline (retrospective) change point detection (Kovács, Bühlmann, Li, & Munk, 2022). Low quantile method has input of sliding time window of past measurements, making it online change point detection method and enabling its use for real-time applications. Low quantile method to detect and compensate indoor CO_2 baseline can also eliminate low outlier data. Thus, the quantile method was selected for CO_2 baseline correction. In the following subsection, parameters of the method are discussed.

Figure 2. *Example of measured CO_2 concentration with an outlier and baseline identified based on minimum and quantile values*



3.2 In depth analysis of parameters of the quantile method

Consider a time series

$$X = (x_t;\ t \in T),$$

where T is the index set. Consider a window of size w and let

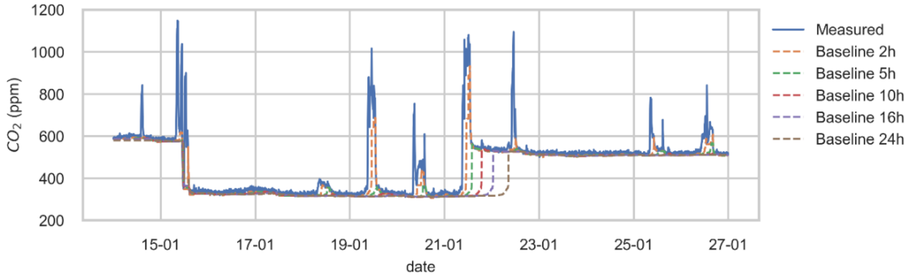
$$X^w = (X_t, \dots X_{t+w}),\ t \in T$$

The formal definition of quantile is given by Hyndman & Fan (Hyndman & Fan, 1996) as

$$Q(p) = F^{-1}(p) = \inf\{x: F(x) \geq p\}\ ,\ 0 < p \leq 1,$$

where $F(x)$ is the distribution of data $x \in X^w$ and p is the quantile.

Figure 3. *CO_2 baseline estimation with different sliding time window length*



There exists multiple definition of sample quantiles (Hyndman & Fan, 1996) in regards of interpolating midpoints. We tried continuous methods where the value is from the real datapoints and not created. Experimenting on collected data and synthesized scenarios with injected outlier data we found that dividing the distribution into 100 partitions was enough for our purposes, so the quantiles that we used were percentiles. Specifically we were interested in 1st and 2nd percentile and calculating baseline as shown in the following.

In $m = p_1 \cdot w$ sample in the ordered statistics $\overline{X}^w$ $of\ X^w$, if m is an integer, we take the m-th value in the set $\overline{X}^w$. If m is not an integer, we use the nearest integer value. More formally:

$$\overline{m} = \begin{cases} m & if\ \ m \in \mathbb{Z} \\ [m] & if\ \ m \notin \mathbb{Z} \end{cases},$$

where [.] notes integer rounding function. With more outlier data ceiling function instead of rounding gave also good results. 1st percentile of described method gave the most precise results when there were no or up to 1% outlier in used time window. In our collected data outlier amount did never exceed 1% making it safe to use 1st percentile. In synthetic scenarios where we had more outlier data then 2nd percentile of the described method gave better results as 1st percentile was trapped by outlier data.

For CO_2 baseline detection we use sliding window of past measurement data. After detecting baseline we corrected the data using following operation:

$$c_t = v_t - b_t + R$$

Where c is corrected value in time t, v_t is measured value, b_t is baseline value and R is agreed baseline value constant (400 ppm). It should be noted that due to the baseline algorithm's online nature, it introduces latency that can be seen in correcting values abovementioned way. Further processing steps to remove outlier data may be necessary. The base value was assumed constant, because the indoor air quality assessment methods provide threshold values for CO_2 concentration excess over outdoor concentration, which is roughly 400 ppm. Ideally, the measured outdoor CO_2 concentration could be used as the base value, but this is difficult to realize as these are not installed in buildings and they should be reliable.

Figure 3 shows a CO_2 level in a room during time period where sensor measurements are rapidly shifting down more than 200ppm and up again after approximately a week. There are dashed lines estimating baseline with different sliding time window lengths. Dashed lines on upper part of the figure show detected baseline and lower part corrected values. It can be seen from the figure that smaller time window causes much noise trying to move baseline even when it should not move. Due to the algorithm's design baseline moving up has latency of the time window size. Longer time windows are more stable but compensating baseline shift takes longer time. On our data the satisfactory time window was 10 hours or more depending on requirement of responsiveness of data.

The proposed method is online, can be used in real-time applications and is utilising sequential historical measurements in sliding time window. The proposed method is usable for baseline detection and does not correct the outlier values. Further smoothing peak values can be done as a post-processing step. The compensation of baseline moving to higher level is delayed and the delay is dependent on used sliding time window size while the shorter time window is less stable.

4. CO_2 concentration correction results

4.1 Typical unexpected CO_2 patterns and the corrections

Typical unexpected patterns described in Section 2.3 and corrected CO_2 values are presented in Figures 4-7 together with data corrected with the developed algorithm. Figure 4 illustrates two outliers, where two initially logged CO_2 values near 0 ppm occurred within two weeks and the corrected CO_2 values. Although, the algorithm presented here does not remove outliers, they did not influence the base level detection. Further development is needed for automated removal of outliers from processed data.

Figure 4. *Initial CO_2 concentration values with outliers and corrected data*

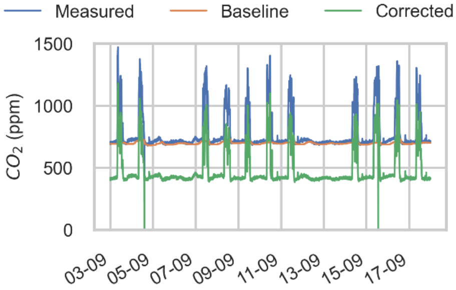


Figure 5. *Initial CO_2 concentration with incorrect baseline value and auto-calibration and corrected data*

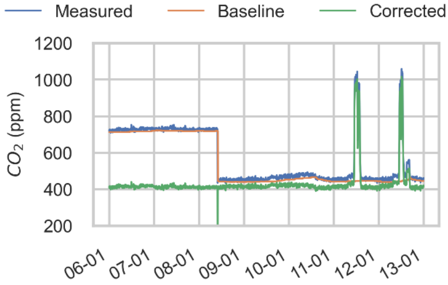


Figure 6. *Initial CO_2 concentration with potentially inadequate sensor placement and corrected data*

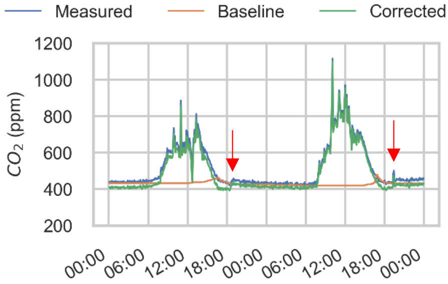


Figure 7. *Initial CO_2 concentration and temperature with CO_2 temperature dependency*

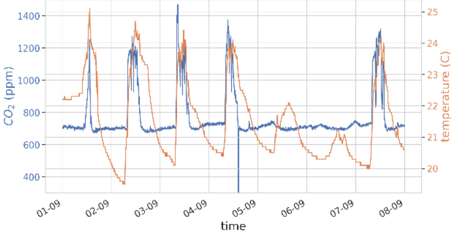


Figure 5 illustrates logged and corrected CO_2 values from a room, where the initial baseline of CO_2 concentration was 700 ppm, which was auto-calibrated to 450 ppm by the measurement device. The corrected values have baseline value 400 ppm.

Figure 6 illustrates logged and corrected CO_2 values from a room, where every school day exponential CO_2 value decay is seen at the presumable end of room occupancy. However, there occurred slight increase in logged value around 19 p.m. on each school day, which then continued to gradually decrease. Our hypothesis is that due to the impact of supply air jet on the sensor, the logged value is lower than the average CO_2 concentration in the room when ventilation system operates. This would mean that the IAQ assessment and ventilation control is inadequate. We have observed this behaviour in other cases also and further field studies and data analysis is needed to potentially develop a method for data-driven identification of such inadequate sensor placement and potential correction of measured CO_2 values. This figure also illustrates that sliding time window length of 10 hours was slightly too short and further development is needed to automatically choose an optimal window length.

7 illustrates logged and corrected CO_2 values and room temperature from a room, where temperature fluctuated significantly between 20 and 24 °C presumably due to solar and internal heat gains and temperature reduction during unoccupied hours. It can be observed that during unoccupied hours, the CO_2 values are negatively correlated with the air temperature. Potential temperature dependency of CO_2 sensors has been identified by Mylonas et al. in laboratory experiments (Mylonas et al., 2019). Further field studies and data analysis is needed to potentially develop a method for data-driven identification of such behaviour and potential correction of measured CO_2 values.

4.2 Qualitative assessment of CO_2 value correction

The performance of used CO_2 base value correction algorithm was assessed qualitatively by comparing initially logged and corrected CO_2 level concentrations. Figure 8 illustrates the respective CO_2 concentrations during 5 weeks in all rooms of the school. The initially logged CO_2 concentration outside expected occupied hours varied constantly over 100 ppm and were typically above 400 ppm. This highlights the need for further post-processing of logged data of the installed sensors despite the in-built auto-calibration function. The corrected values reflect the expected behaviour described in Section 2.2. However, there still were some outliers with values significantly below 400 ppm due to the chosen quantile method, which should be removed by further developing the data-processing methods before applied for IAQ assessment or ventilation control.

The performance of the base level correction method was additionally assessed by comparing the duration curves of initially logged and corrected data. An example based on a classroom is provided in Figure 9 and the results of all rooms with CO_2 sensors are given in Figure 10. Visually, the largest difference appears during periods when the CO_2 values are near outdoor level.

However, to quantify the impact of the baseline level correction algorithm during occupied period, the initially logged and corrected values were corrected by comparing them to the recommendations in the REHVA COVID-19 guidance. In the beginning of 2023, the guidance recommends 550 ppm as setpoint for demand-controlled ventilation systems (REHVA, 2021). In 2020, the suggestion was: *"During an epidemic it is recommended to temporarily change the default settings of the traffic light indicator so that the yellow/orange light (or warning) is set to 800 ppm and the red light (or alarm) up to 1000 ppm in order trigger prompt action to achieve sufficient ventilation even in situations with reduced occupancy."* (REHVA, 2020).

Figure 8. *Example of CO_2 concentration of all rooms during 5 weeks with initially logged (above) and corrected (below) baseline values*

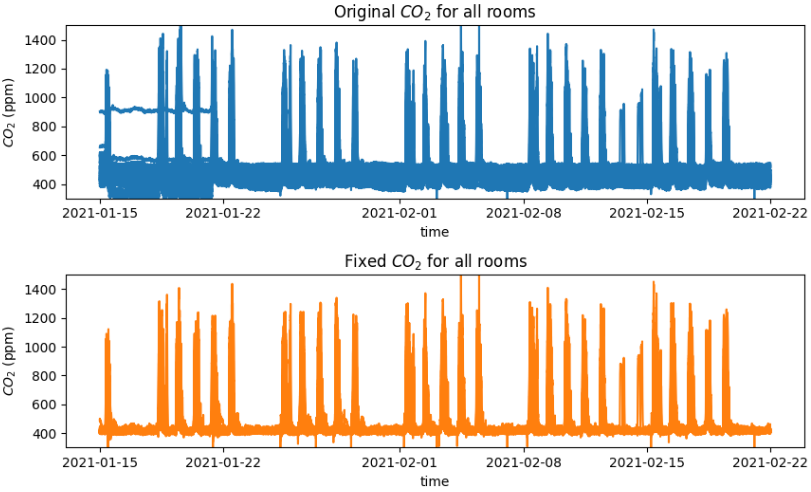


Figure 9. *Example of cumulative distribution of initially logged and corrected CO_2 concentration in one classroom during 6-month period.*

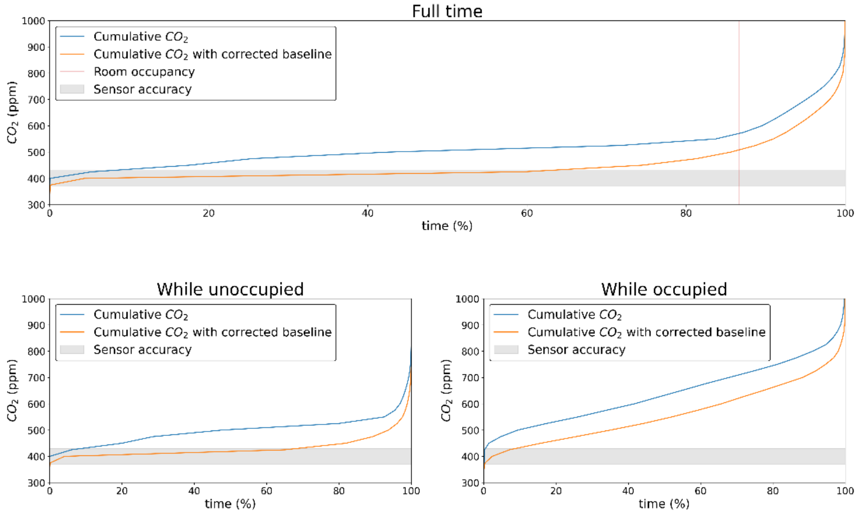


Table 1 shows the impact of baseline level correction of comparing logged values against different CO_2 concentration levels. The pandemic situation recommended ventilation control setpoint 550 ppm for was exceeded during 89.1 and 74.4% of occupied period with initially logged and corrected data respectively. Therefore, the impact of baseline level correction would have been significant if the school had demand-controlled ventilation and the recommendations followed. The differences in exceedances of the 800 and 1000 ppm were even more significant. As initially logged CO_2 concentrations were generally higher than the corrected values, then the initially logged values would have prompted unnecessary disturbances in school work due to opening windows during classes or even stopping schoolwork due to increased infection-risk. Thus, the practical implication of CO_2 concentration level correction needs to be further studied to develop the more robust methods for IAQ assessment.

Figure 10. *Cumulative CO_2 of all rooms in the same timeframe as previous figure*

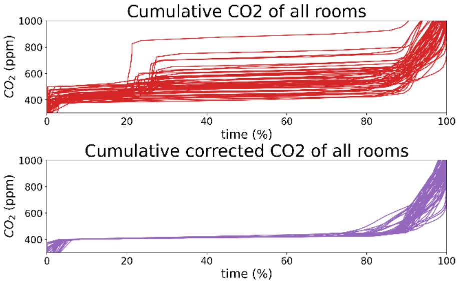


Table 1. *The impact of baseline level correction of comparing logged values against different CO_2 concentration levels*

	Time of occupied period above respective CO_2 level, %		
	550 ppm	800 ppm	1000 ppm
Initial	89.1	46.2	18.2
Corrected	74.4	27.7	5.9

4.3. Limitations and future work

The subsequently presented method for CO_2 base level correction was developed based on measured data from an adequately ventilated school. It is unknown, how would the method perform in poorly ventilated or constantly operated rooms, where the CO_2 concentrations might not reach the base level. Therefore, the proposed sliding time window length of 10 hours needs to be carefully assessed when applying the method in building with different ventilation principles of duration of occupancy. Additionally, with worse quality data that contains more outliers, 2% or higher percentile values should be used for adequate results.

Additionally, no reference measurements with calibrated CO_2 sensors were conducted and it is unknown, how adequate were the CO_2 levels during occupied hours of both initially logged and corrected data. Hypothetically, some logged values depended on room temperature and unexplainable jumps in logged values were observed. Further studies are needed to investigate the possibilities of further development of data-driven assessment of CO_2 concentration.

The currently method was developed for correction of CO_2 concentrations in real-time without knowing future values. Therefore, the current implementation is more effective capturing the downward jumps of CO_2 base level. The baseline correction method can be implemented also for regular IAQ assessment e.g. with weekly or monthly frequency and for such application the method should be developed further to also capture effectively the upward jumps, which is currently delayed.

5. Conclusion

Indoor air quality assessment based on CO_2 sensors is more frequently used, however the logged data is often flawed and needs to be further processed for adequate application in ventilation control or indoor climate assessment. This study developed algorithm for CO_2 base level correction in real-time based on the assumption that CO_2 concentration reaches 400 ppm. The algorithm was developed based on CO_2 data logged during 6 months from 56 rooms of a well-ventilated school building in Estonia. The algorithm performance was assessed qualitatively.

The algorithm used percentile values from a sliding time window to identify the base level and subsequently the data was corrected so that the base level would be 400 ppm. The best compromise between accuracy and delay in data correction was reached by using 1% percentile values in a 10 hour sliding time window. The impact of the algorithm was significant when comparing the initially logged and corrected values against CO_2 concentration thresholds 550, 800 and 1000 ppm.

The developed method needs to be further tested based on data from inadequately ventilated buildings, building with different occupancy patterns and sensors that might provide worse quality data that contains more outliers. Additionally, the post-processed CO_2 concentration needs comparison with more detailed measurements with calibrated sensors for further validation and development. Finally, the current implementation of the algorithm effectively captures downward shifts in the logged data, but future development is needed to capture upward shifts with smaller delay.

Acknowledgements

This work has been supported by the Estonian Ministry of Education and Research, European Regional Fund (grant 2014-2020.4.01.20-0289), the Estonian Centre of Excellence in Zero Energy and Resource Efficient Smart Buildings and Districts, ZEBE (grant 2014- 2020.4.01.15- 0016) funded by the European Regional Development Fund, by the European Commission through the H2020 project Finest Twins (grant No. 856602), the Estonian Research Council grant (PSG409) and by Estonian Centre of Excellence in IT (EXCITE), funded by European Regional Development Fund.

References

European Commission. (2021). *Proposal for a Directive of the European Parliament and of the Council on the energy performance of buildings (recast)*. Retrieved from https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A52021PC0802

Hyndman, R. J., & Fan, Y. (1996). Sample Quantiles in Statistical Packages. *The American Statistician*, *50*(4), 361. https://doi.org/10.2307/2684934

Indoor Air Quality Monitor. (2020). *Smart Temp Australia*. Retrieved from https://smarttemp.com.au/wp-content/uploads/2021/10/SMT-IAQ3-manual.pdf

Kovács, S., Bühlmann, P., Li, H., & Munk, A. (2022). Seeded binary segmentation: a general methodology for fast and optimal changepoint detection. *Biometrika*. https://doi.org/10.1093/biomet/asac052

Mylonas, A., Kazanci, O. B., Andersen, R. K., & Olesen, B. W. (2019). Capabilities and limitations of wireless CO2, temperature and relative humidity sensors. *Building and Environment*, *154*, 362–374. https://doi.org/10.1016/j.buildenv.2019.03.012

Pedersen, T. H., Nielsen, K. U., & Petersen, S. (2017). Method for room occupancy detection based on trajectory of indoor climate sensor data. *Building and Environment*, *115*, 147–156. https://doi.org/10.1016/j.buildenv.2017.01.023

REHVA. (2020). *REHVA COVID-19 guidance document*. Retrieved from https://www.rehva.eu/fileadmin/user_upload/REHVA_COVID-19_guidance_document_V3_03082020.pdf

REHVA. (2021). *REHVA COVID-19 guidance, version 4.1*. Retrieved from https://www.rehva.eu/fileadmin/user_upload/REHVA_COVID-19_guidance_document_V4.1_15042021.pdf

Thompson, J. (2014). *An Empirical Evaluation of Denoising Techniques for Streaming Data*. https://doi.org/10.2172/1165751

Appendix 4

IV

L. Vihman and J. Raik. Adaptive Kalman Filter Based Data Aggregation in Fault-Resilient Underwater Sensor Networks. In *2023 24th Int. Conf. Digit. Signal Process.*, pages 1–5. IEEE, jun 2023

Adaptive Kalman Filter Based Data Aggregation in Fault-Resilient Underwater Sensor Networks

1st Lauri Vihman
Department of Computer Systems, Taltech, Estonia

2nd Jaan Raik
Department of Computer Systems, Taltech, Estonia

***Abstract*—Sensor Networks in harsh underwater environments are prone to faults and anomalies that may lead to deteriorated data quality or even failures. This paper proposes a fault-resilient underwater sensor network based on sensor data aggregation by updating the measurement error matrix of an adaptive Kalman filter, where the error matrix is updated using adjusted measured value difference from the predicted value as well as the age of the latest measurement (i.e. latency). A case study on a real-world harbor water flow monitoring use-case shows the advantages of the proposed method. The experiments indicate that the adaptive and difference based Kalman filter aggregation provides for a significantly smoother aggregation in case of high fault rates in sensors' readings when compared to traditional Kalman filter and median value based aggregation techniques.**

I. INTRODUCTION

Data Aggregation methods are widely used to conserve energy and communication bandwidth in sensor networks [1]–[5]. When sensor nodes have limited energy, it is reasonable not to send all the data to the receiving sink. Data aggregation is referred as amongst the most important energy-saving analysis processes [3]. In recent years, research has been done on aggregation methods that are best applicable for underwater environments [1], [2]. In addition, aggregation techniques have been used to address security concerns (compromised nodes), data integrity and redundancy as well as the lifetime of sensor networks [4].

Methods of data aggregation for network communication can be distinguished by structure - flat methods, where all the sensors are playing the same role, clustering-based, tree-based and hybrid methods [2] and by functional approach like aggregation by feedback control, quantile digest, distributed source coding [5].

The Kalman Filter (KF) is one of the most popular mathematical state estimation tools [6] that includes multiple variants and is extensively used in robotics [7]–[11] and combined to fuzzy logic [9], [10] and sensor fusion [7], [9]–[13] techniques.

An Extended Kalman Filter has been applied to estimate gas leaks in pipelines [6] and to overcome signal noise as well as limitations of different kinds of sensors [12].

For sensor data aggregation, Kalman Filters have been used as a security tool to detect false data injection attacks in sensor networks [14] and also for noise elimination from signals with ordered weight averaging [15] and in wireless sensor networks to achieve a distributed consensus [16].

979-8-3503-3959-8/23/$31.00 ©2023 IEEE

While above-mentioned works on Kalman Filters did not address hardware faults, the following works focus also on that aspect. A generic detection and compensation of occurrence of transient and permanent faults was described [13] using Kalman Filter and correspondingly registering T_{Fault} and P_{Fault} variables, respectively, when values were exceeding a defined threshold. However, it relied on a difference between measured and predicted values compared to a defined threshold and did not consider signal latency nor sensor outage. Kalman filter based fault diagnosis and accommodation has been studied in [8] limited to robot wheel actuators and optical encoders. A more recent study using a Kalman Filter for robot localization can overcome faults including sensor outage and data corruption of IMU sensors [11]. However, no sensor signal latency is considered and the prediction weights are linear.

While previous works have applied Kalman filters in fault-tolerant applications to the best of the authors' knowing this is the first work to propose a KF that includes different uncertainty sources and applies the adaptive KF based data aggregation in USNs.

Contributions of this work are as follows:

- Incorporating different sources of sensor uncertainty by including the time series measurements' difference and age/latency uncertainty for adapting a KF to compensate incorrect readings for avmore efficient state prediction.
- Proposing nonlinear, parabolic and sigmoid, sensor uncertainty functions from the residual difference for the latency and difference based Adaptive Kalman techniques, respectively.
- Applying and evaluating the proposed adaptive KF based data aggregation techniques in a harbor Underwater Sensor Network (USN) with extremely unreliable sensor readings.

II. UNDERWATER SENSOR NETWORK APPLICATION

In this Section, we introduce the layered data architecture and installation of the sensor network in the application scenario of a harbor monitoring use case.

A. Sensor Network Installation

The underwater sensor network considered in this paper is for monitoring sea currents in the harbor. The Sensor Nodes $S = \{s_i\}$ of the network are installed to the harbor infrastructure to notify approaching ships about the water

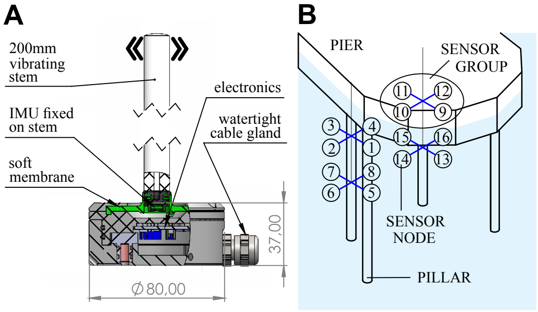


Fig. 1. A) Sensor Node B) Harbor Installation

flow around the piers. The goal is that berthing ships get information about the flow and turbulence from Sensor Nodes installed on the pillars of the pier. The Sensor Nodes are connected to the Sensor Gateway with underwater cables over RS-485 serial communication, thus the configuration of the underwater sensor network is fixed.

The Sensor Node applied in the case study is shown in Figure 1.A and its detailed description is given in [17]. The Sensor Nodes measure the flow magnitude and direction from the stem vibrating in the flow. An IMU (Inertial Measurement Unit) embedded into the Sensor Node calculates the accelerations of the stem in x and y directions. In the current implementation, to estimate the flow magnitude from the IMU data, a 15s series of IMU raw data is transformed with FFT (Fast Fourier Transform) into a frequency domain in 120s intervals and the PSD (Power Spectral Density) is used to find the flow magnitude using calibrations described in [17]. All the Sensor Nodes are wired to the Sensor Gateway in a star network topology.

The Sensor Nodes $\{s_i\}$ are installed around the pillars at two different depths so that at both depths 4 Sensor Nodes are attached around the pillar at 90 degrees angle from each other forming a logical Sensor Group $G = \{g_j\}$. This is necessary because depending on the direction of the flow, the pillar itself always obstructs some of the Sensors Nodes from the flow. Therefore each aggregation of 4 Sensor Nodes (Sensor Group) is used to estimate the flow at each point.

In total the installation has 16 Sensor Nodes, grouped into 4 aggregations (Sensor Groups) of 4 sensors each as shown in Figure 1.B.

B. Layered data model implementation

Table I presents the data-driven layers of the sensor network architecture. There are three data layers - raw, processed and aggregated that correspond to different functionalities that can be loosely mapped to IoT layers as shown in Table I.

TABLE I
DATA LAYERS AND THEIR MAPPING TO IOT LAYERS

Data Layer	**State**	**IoT Layers**
Raw	direct data from sensors	Edge
Processed	pre-processed data from sensors	Edge/Fog
Aggregated	data combined to sensor groups	Cloud

At the **Raw data layer** is data that is measured and transferred by the set of Sensor Nodes $S = \{s_i\}$. The Sensor Nodes measure and send values periodically and the event of receiving a valid test packet at the Raw Data Layer by sensor s_i at a discrete time instance t is $\rho(i,t)$. The raw measurement value $\rho(i,t)$ of the sensor node s_i at time t is a vector containing values such as Inertial Measurement Unit's XYZ positions, temperature and pressure.

All the valid $\rho(i,t)$ measurement values received during a time interval T, containing n time instances, are pushed into the raw data queue of length n, which acts as a First-In-First-Out (FIFO) data buffer $F = \{f_1, f_2, ..., f_n\}$, where $f_1, ..., f_n$ represent n latest valid $\rho(i,t)$ measurement values. There is a separate queue F_i for each sensor node s_i.

Processed data layer is used to obtain the processed data. At the predefined time intervals the data in F are processed by a signal processing function Ψ to obtain the processed data value of $\sigma(i,t)$, i.e. $\sigma(i,t) = \Psi(F)$

Initially Ψ transforms the raw values $\rho(i,t)$ from F_i into an intermediate processed values $\sigma'(i,t)$. $\sigma'(i,t)$ contains values that are in a linear relation to quantities required by the end user.

Next, the processing function Ψ converts values from $\sigma'(i,t)$, using predefined calibration and offset constants that are specific for the installation and concrete sensor instance, to quantities required by the end user of the harbor water flow monitoring application (e.g. velocity).

Aggregated data is generated for the Sensor Groups $G = \{g_j\}$ from the processed data σ of the Sensor Nodes $\{s_i\}$. The purpose of the aggregation process is to provide transition from single Sensor Nodes' data to Sensor Groups' data. Subsequently, the aggregated value $\alpha(j,t)$ of the sensor group g_j at Aggregated data layer is calculated by an aggregation function Ω on converted values $\sigma(i,t)$ of the sensor nodes s_i that belong to the sensor group g_j, i.e. $\alpha(j,t) = \Omega(\{\sigma(i,t) | s_i \in g_j\})$.

III. FAULT-RESILIENT DATA AGGREGATION

In the current Section, we propose an implementation of the data aggregation function Ω introduced in the previous section. Input values for the aggregation function are the processed values $\sigma(i,t)$ and output values are $\alpha(j,t)$, respectively.

A. Adaptive Kalman filter for data aggregation

Kalman Filter (KF) is a widely used technique for data analysis, such as filtering, smoothing, initialization, forecasting, assimilation and aggregation [18]. In our case, due to physical limitations, harsh underwater environment and possibility of occurrence of both, persistent and intermittent faults, the sensors vary from correct measurements. To cope with that issue, we rely on a Sensor Group g_j (see II-B) to generate data fusion for univariate measurements - that is multiple sensors simultaneously measure similar physical entity. We are applying KF in the Aggregation Data layer (see II-B) after an initial signal processing is done.

We are using adaptive KF for data fusion to compute aggregated data and get the filtered estimate that is more reliable than the sources. Sensor values are read and transformed to velocity values simultaneously after a predefined constant time interval. To calculate aggregated values of the 4-sensor groups, adaptive KF is used. The KF is implemented as follows:

$$\begin{aligned} X_{t|t-1} &= AX_{t-1|t-1} \\ V_t &= Y_t - X_{t|t-1}H \\ P_{t|t-1} &= AP_{t-1|t-1}A^T + Q \\ S_t &= HP_{t|t-1}H^T + R_t \\ K_t &= P_{t|t-1}H^T S_t^{-1} \\ X_{t|t} &= X_{t|t-1} + K_t V_t \\ P_{t|t} &= P_{t|t-1} - K_t S_t K_t^T \end{aligned}$$

where $t \in N$ is discrete time, $Y \in \mathbb{R}^4$ is the measurement vector containing σ_i values of the current sensor group, $X \in \mathbb{R}$ is the state estimate scalar, $Q \in \mathbb{R}$ is process noise scalar constant, $H \in \mathbb{R}^4$ is the observation vector constant, $V \in \mathbb{R}^4$ is the calculated innovation residual vector, $A \in \mathbb{R}$ is state matrix constant, $P \in \mathbb{R}$ is the updated estimate covariance scalar, $R \in \mathbb{R}^{4\times4}$ is the sensor uncertainty co-variance matrix, $S \in \mathbb{R}^{4\times4}$ is the innovation co-variance matrix and $K \in \mathbb{R}^4$ is the Kalman gain vector.

The sensor uncertainty covariance matrix R is updated using L_i and D_i weights (see III-B) in different configurations as explained next.

In case of *Kalman Static*, sensor uncertainty R is not updated during the filter's execution.

In case of *Kalman Difference* aggregation, sensor uncertainty matrix R is set to $n(D_i) \times J_4$, where J_4 is the unit matrix and n is the normalization function explained in III-B.

In case of *Kalman Latency* aggregation, sensor uncertainty R is set to $n(L_i) \times J_4$.

In case of *Kalman Adaptive* aggregation, sensor uncertainty R is set to $(n(L_i) + n(D_i)) \times J_4$.

The output value of the aggregation function Ω, i.e. α_j of the sensor group g_j at time t receives its value from $X_{t|t}$.

The calculation is iterative over time t and for aggregation, the values X_t are used. To make KF adaptive, the covariance matrix R_t is externally updated increasing sensors' uncertainties with increasing difference from aggregated X_t value and measurement latency, indicating outliers and probable sensor faults.

B. Uncertainty from the difference of measurement and estimation

When detecting outlier values based on the difference between estimation and measurement, we argue that this should not be linear - small differences in velocity should be proportionally more tolerated than larger differences. The uncertainty value based on the difference does not have to have an upper limit, since the type of the sensors used that have a defined measurement range - frequencies beyond this range are not reported and thus cannot affect the uncertainty value of the sensor. For adaptive weights we applied the following function $D_i = c(\sigma(i) - x)^2$, where $\sigma(i)$ is the processed value of the sensor s_i, x is the predicted value of KF, thus $\sigma(i) - x$ is the residual, and c is a constant 1.5 that was chosen empirically. The normalized graph of the function is shown in Figure 2 right.

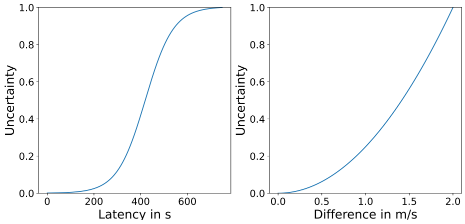


Fig. 2. Normalized sensor uncertainty from measurement and prediction difference (residual) and the latest measurement latency

For uncertainty caused by measurement latency we applied the following sigmoid function. $L_i = \frac{1}{1+e^{\frac{-t_i}{m}+h}}$, where m = 60 and h = 7 and t_i is the latency of the latest valid measurement of the i-th sensor s_i in seconds. Normalized uncertainty values based on latency are presented in Figure 2 left.

The normalization is performed to be able to compare the uncertainty function values and give them weights to calculate the sensor uncertainty co-variance matrix using the following normalization function $n(v) = \frac{v - min(V)}{max(V) - min(V)}$, where v is a value from a vector of values V to be normalized.

IV. EXPERIMENTS

Experiments in two different underwater environments were conducted. The Flow Obstuction experiment was a short time experiment that took place in freshwater in a river on February 2, 2023. For this experiment, in addition to the hydromast sensors, ADV (Acoustic Doppler Velocimeter) measurements were also used as reference values. The sensor network was installed to a river bed (See IV-A) and the water flow was manually disturbed and interfered.

The Harbor Experiment was a long-time experiment active from April to August 2020. The sensor network was installed into sea water by a harbor for measuring underwater currents (See II-A). For this experiment we did not have a reference device. The sensors were not disturbed nor interfered manually, the collected data was naturally occurring. Most of the time during that period the water flow was too slow to be measurable with sensors due to non-windy weather conditions. However, there were a couple of time intervals with a stronger water movement.

A. *Flow obstruction experiment*

The hydromast sensors were attached on a metal bar at 20cm intervals. Perpendicular to the centre of the hydromast was another bar with attached ADV (Acoustic Doppler Velocimeter, Nortek Vectrino Profiler) approximately 50cm from the hydromast metal bar. The construction was installed to a river bottom around 1m depth with the ADV facing the flow and hydromasts side by side behind it. The order of the hydromasts from the shore was H24, H25, H26, H27. The unobstructed water velocity appeared similar at all hydromasts. The hydromast offset coefficients are calibrated after installation to correspond to ADV beams mean value magnitude. The velocity is calculated using the magnitude of median x and y axis angles of 1 sec time frame of 50hz measurements.

Figure 3 shows the obstruction experiment.

- Sensor measurements are shown as dots. Aggregate different Kalman filters are shown as lines.
- A human was obstructing the flow by standing in the water for every hydromast for 30 sec in the following order - 1st H24, 2nd H25, 3rd H26, 4th H27
- It can be seen that the obstruction changed the hydromast angles correspondingly as the dots representing single measurements move downwards at specific times.
- H25 obstruction is less clear, but happened also while obstructing H24 and H26, thus standing near the 1st and 3rd hydromast obstructed the flow also at the 2nd hydromast.

It can also be seen from Fig. 3 that the sensors' water flow measurement is consistent and adequately reacts to changes in the water flow. From the Figure, it can be seen that Kalman difference is the most optimal aggregation method for this case, as it is accurately filtering out disturbances at individual sensors. It was followed by median and Kalman adaptive aggregation. However, Kalman latency and Kalman static were far more tolerant to disturbances at individual sensors.

B. *Harbor experiment*

Finally an experiment was carried out on naturally occurring data from the actual use case. From the data collected during the five-month period we selected an interval on May 7th, 2020 where all of the sensors in the sensor group were active and there was enough flow to measure velocity and direction. The flow is in a measurable range from approximately 6:30 to 20:00 when it begins to fade. The day is characteristic for representing the harshness of the environment as there are multiple gaps and outliers in the readings and one of the sensors (H3) stops providing new measurements around 13:20.

In this harsh, real-world environment Kalman difference and Kalman adaptive are performing equally efficiently, while the aggregation provided by Median and Kalman latency methods is far too unstable. The weakest performance is obtained by Kalman static that is consistently overestimating the water current flow.

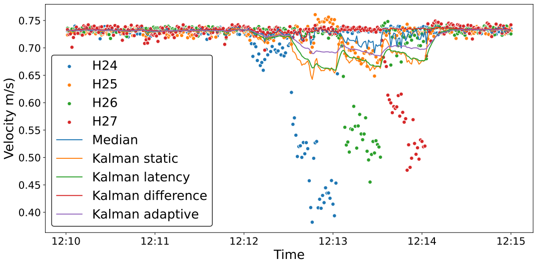


Fig. 3. Comparison of the aggregation methods in the obstruction experiment

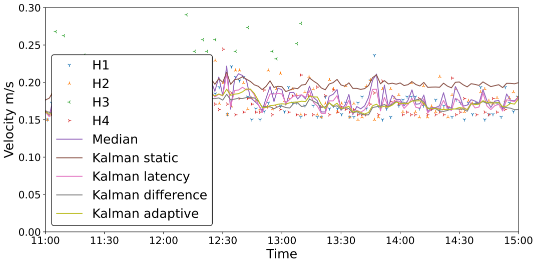


Fig. 4. Aggregation methods and processed values for velocity measurement in the harbor experiment

As the result of the experiments, the most robust and stable aggregation performance was achieved by the Kalman difference method that took into account the difference of the measurement value at the sensor from the estimated value. This method performed well in a stable current as well as in case of disturbances and also in very harsh conditions, where there were gaps and outliers.

Kalman adaptive was slightly less accurate with faulty sensor reading but became more robust and equal to Kalman difference in case of more frequent gaps in readings. It might become the preferred option when conditions are extremely harsh and become even more dominated by gaps.

V. CONCLUSION

A fault-resilient underwater sensor network based on sensor data aggregation by updating the measurement error matrix of an adaptive Kalman filter was proposed. A case study on a real-world harbor water flow monitoring use-case showed that the adaptive and difference based technique allowed for a significantly smoother aggregation in case of high fault rates in sensors' readings when compared to traditional Kalman filter and median value based aggregation techniques.

ACKNOWLEDGMENT

This work has received funding from the European Commission's Horizon 2020 Research and Innovation programme under grant agreement No 101037643 and by Estonian Centre of Research Excellence EXCITE.

REFERENCES

[1] N. Goyal, M. Dave, and A. K. Verma, “Data aggregation in underwater wireless sensor network: Recent approaches and issues,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 31, pp. 275–286, jul 2019.

[2] S. Abbasian Dehkordi, K. Farajzadeh, J. Rezazadeh, R. Farahbakhsh, K. Sandrasegaran, and M. Abbasian Dehkordi, “A survey on data aggregation techniques in IoT sensor networks,” *Wirel. Networks*, vol. 26, no. 2, pp. 1243–1263, 2020.

[3] S. A. Abdulzahra and A. K. M. Al-Qurabat, “Data Aggregation Mechanisms in Wireless Sensor Networks of IoT: A Survey,” *Int. J. Comput. Digit. Syst.*, vol. 13, pp. 1–15, jan 2023.

[4] K. Gulati, R. S. Kumar Boddu, D. Kapila, S. L. Bangare, N. Chandnani, and G. Saravanan, “A review paper on wireless sensor network techniques in Internet of Things (IoT),” *Mater. Today Proc.*, vol. 51, pp. 161–165, 2022.

[5] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, “In-network aggregation techniques for wireless sensor networks: a survey,” *IEEE Wirel. Commun.*, vol. 14, pp. 70–87, apr 2007.

[6] R. Doshmanziari, H. Khaloozadeh, and A. Nikoofard, “Gas pipeline leakage detection based on sensor fusion under model-based fault detection framework,” *J. Pet. Sci. Eng.*, vol. 184, no. July 2019, 2020.

[7] T. Huang, H. Jiang, Z. Zou, L. Ye, and K. Song, “An integrated adaptive Kalman filter for high-speed UAVs,” *Appl. Sci.*, vol. 9, no. 9, 2019.

[8] S. Mellah, G. Graton, E. M. E. Adell, M. Ouladsine, and A. Planchais, “Mobile robot additive fault diagnosis and accommodation,” *2019 8th Int. Conf. Syst. Control. ICSC 2019*, pp. 241–246, 2019.

[9] J. Z. Sasiadek and P. Hartana, “Sensor data fusion using Kalman filter,” *Proc. 3rd Int. Conf. Inf. Fusion, FUSION 2000*, vol. 2, pp. 19–25, 2000.

[10] P. J. Escamilla-Ambrosio and N. Mort, “Hybrid Kalman Filter-Fuzzy Logic Adaptive Multisensor Data Fusion Architectures,” *Proc. IEEE Conf. Decis. Control*, vol. 5, no. December, pp. 5215–5220, 2003.

[11] M. Kheirandish, E. A. Yazdi, H. Mohammadi, and M. Mohammadi, “A fault-tolerant sensor fusion in mobile robots using multiple model Kalman filters,” *Rob. Auton. Syst.*, vol. 161, 2023.

[12] E. Foxlin, “Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter,” *Proc. - Virtual Real. Annu. Int. Symp.*, pp. 185–194, 1996.

[13] P. J. Escamilla-Ambrosio and N. Mort, “Multi-sensor data fusion architecture based on adaptive Kalman filters and fuzzy logic performance assessment,” *Proc. 5th Int. Conf. Inf. Fusion, FUSION 2002*, vol. 2, no. 3, pp. 1542–1549, 2002.

[14] K. Miao, W. A. Zhang, and X. Qiu, “An adaptive unscented Kalman filter approach to secure state estimation for wireless sensor networks,” *Asian J. Control*, vol. 25, no. 1, pp. 629–636, 2023.

[15] M. H. Basiri, N. L. Azad, and S. Fischmeister, “Distributed time-varying kalman filter design and estimation over wireless sensor networks using OWA sensor fusion technique,” *2020 28th Mediterr. Conf. Control Autom. MED 2020*, pp. 325–330, 2020.

[16] J. Qin, J. Wang, L. Shi, and Y. Kang, “Randomized Consensus-Based Distributed Kalman Filtering over Wireless Sensor Networks,” *IEEE Trans. Automat. Contr.*, vol. 66, no. 8, pp. 3794–3801, 2021.

[17] A. Ristolainen, J. A. Tuhtan, and M. Kruusmaa, “Continuous, near-bed current velocity estimation using pressure and inertial sensing,” *IEEE Sens. J.*, vol. PP, no. c, pp. 1–1, 2019.

[18] J. Durbin and S. J. Koopman, *Time Series Analysis by State Space Methods*. Oxford University Press, may 2012.

Appendix 5

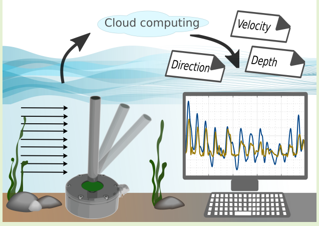
V

M. Egerer, A. Ristolainen, L. Piho, L. Vihman, and M. Kruusmaa. Hall effect sensor-based low-cost flow monitoring device: Design and validation. *IEEE Sensors Journal*, pages 1–1, 2024

Hall Effect Sensor-Based Low-Cost Flow Monitoring Device: Design and Validation

Margit Egerer, Asko Ristolainen, Laura Piho, Lauri Vihman, Maarja Kruusmaa

***Abstract*—Monitoring and assessment of coastal and river velocities plays a key role in both scientific and industry applications. Field measurements are key for decision making for resource management and protection, as well as for validation of numerical models and climate change studies. In this paper, a Hall effect sensor-based cost effective novel device is proposed for measuring velocities and direction of near-bed currents, and water-level. This device, called the Hydromast, provides instantaneous measurements in real-world conditions, and is equipped with communication capabilities to allow near real-time data transfer and monitoring. The validation of the device is performed in real-world steady and unsteady flow conditions. Within the device measurement range it is shown that the root-mean-square-error of the time averaged flow measurements is under 0.1m/s.**



***Index Terms*—Water current measurements, remote data transmission, flow direction, flow velocity, near-bed measurements**

I. INTRODUCTION

Measuring flow velocity in the field conditions plays a significant role in many scientific and industry applications. For example in hydrological studies [1], sediment transport investigations [3]–[5], the determination of aquatic habitats [6], [7] in rivers, estuaries and coastal waters, and flood warning systems [8]. Near-bed velocity estimates are key metrics in sediment transport and river habitat studies [9], [10].

Popular in-situ field measurement devices for flow velocity are propeller velocity flowmeters, acoustic Doppler velocimeters (ADV) and acoustic Doppler current profilers (ADCP). In addition, there have been efforts to develop remote sensing methods, from the use of radars for estimating surface velocities to satellite imagery for river discharge estimation [11], [12]. However, the aforementioned methods have either good temporal resolution or good spatial resolution, not both. The acoustic measuring devices (ADV and ADCP) have a good temporal resolution of $50\,\mathrm{Hz}$, whereas the spatial resolution depends on the number of devices. Due to the high cost, in general, not many of these devices are used together, especially over long periods of time and in extreme environments where the chances of recovering the instruments goes down. Satellite models e.g., Planet Labs satelite SkySat [13], [14] has good spatial resolution (about˘$0.5\,\mathrm{m}$). However, the temporal resolution of SkySat is 4-5 days [13], [14].

Further problems can arise when measuring near-seabed or near-surface velocity, as local wake features can affect the measurements near the bed and air entertainment, secondary currents, and velocity tip effect can influence near surface measurements [15]. In addition, remote sensing methods are restricted to estimating water surface velocity [16]–[18]. In [19], it is shown that to obtain reliable average velocities in a flow affected by natural turbulence and instrument noise, sampling duration of 90 and $150\,\mathrm{s}$ are found to be sufficient for ADV and ADCP, respectively. For long-term behavior and large scale spectral analysis, when many sources of flow variability are present, longer sampling duration is needed. This means that for reliable estimation of near-surface velocities ADV and ADCP temporal resolution also decreases.

Despite the challenges, for many applications, continuous instantaneous flow measurements are extremely important. Sediment motion depends on momentary flow features [20], flow type characterization [21] and feature detection in flows (e.g., ship detection [22]) require high temporal resolution, studying stresses on submerged structural elements benefit from long-term continuous monitoring [23]. Therefore, there is a need for cost effective methods providing continuous, reliable and distributed data of the near-bed velocities.

Motivated by the changing hydrological conditions imposed by climate crisis and the need for near-bed continuous measurements, a method for in-situ observations of flow velocity was proposed by Ristolainen et.al [24]. A bimodal flow sensing device using accelerometers was designed and used to automatically classify river hydromorphology [25], and new methods for near-bed velocity measurement were developed [26]. Unfortunately, this device, the original Hydromast, had several drawbacks: it had a limited flow velocity range; it

Date on which you submitted your paper for review. This work was supported by the European Union through European Social Fund project "ICT programme", H2020 project ILIAD (grant agreement No 101037643), H2020 project LAKHSMI (grant agreement No 635568.) and from the Estonian Research Council funded project SolidShore (grant agreement EMP480).

The authors are with the Centre of Biorobotics, Department of Computer Systems, School of Information Technologies, Tallinn University of Technology, 12618 Tallinn, Estonia (e-mail: margit.egerer@taltech.ee).

could only estimate the average mean flow velocity; and it was not capable of real-time data output. The lower end of the measurement range was too high for many near-bed applications, allowing only measurements in constantly fast flows. Additionally, no measurements were possible in wave-driven coastal applications, with regularly changing flow direction and only local storage was possible, allowing data processing and output only after the recovery of the devices.

To address the described limitations, a novel design of the Hydromast based on a Hall effect sensor has been proposed and a cloud based communication set-up has been designed, granting near real-time (latency couple of seconds) data transfer and observations by the user. The upgraded device, from now on referred to as the Hydromast, uses magnetic field sensing at the base to instantaneously detect the exact tilt and direction of the sensing element. This allows simpler and faster estimation of flow velocity and additionally allows to measure the direction of the flow. Changing length of the sensing element allows varying the velocity measurement range according to the application. Moreover, the new design allows for instantaneous flow estimates, and opens up a new application range of unsteady flow measurements in rapidly changing environments. The device has been validated in various flow fields together with a baseline measurement for determining the behavior, robustness and durability in real-life environments, and demonstrating possible applications. This leads to an accurate, robust, and cost efficient way to measure continuous near-bed current velocity.

The organization of this work is as follows; Section II introduces the working principle, device design and upgrades. Section III describes the calibration methods and results. Section IV provides the description and results of the field validations in both steady and unsteady flow. Sections V and VI discuss the results and performance of the device presented in the paper and bring out the potential applications, strengths and weaknesses of continuous instantaneous velocity estimation with the Hydromast.

II. HYDROMAST DESIGN AND WORKING PRINCIPLE

The Hydromast is inspired by the neuromast, which is the major unit of functionality of the biological lateral line. A neuromast, being a mechanoreceptive organ of fish, is responsible for the sensing of mechanical changes in the surrounding flow field [2]. The Hydromast consists of a rigid mast that is fixed to the base with a flexible membrane, resembling an up-scaled version of a neuromast. The bulk flow velocity over the mast generates vortex induced vibrations (VIV) of the mast which dominate over the random forcing due to turbulent flow conditions.

A. Hydromast Design

In the previous Hydromast designs by [24] and [26] the mast motion was recorded with a micromechanical inertial measuring unit (IMU). Our upgraded Hydromast design uses a 3D Hall effect sensor (TLV493D-A1B6, Infineon Technologies AG) to track the mast movement. The Hall effect sensor, located in the Hydromast base, detects the strength of the magnetic field from a 5x5 mm neodymium cylindrical magnet installed at the base of the mast inside the flexible membrane. A main benefit of implementing the magnetic field sensing is the instantaneous position estimation of the mast for instantaneous tilt angle and direction measurements. Also, the contactless sensing of magnetic fields improves the robustness of the device, as no cables are connected to and affecting the vibrating mast. This also simplifies and reduces the cost of the manufacturing. Similarly to the previous design, the Hydromast can be equipped with an external IMU (MinIMU-9, Pololu Corporation) to detect the installation angle and device base movements in unsupervised installations onto sea bead.

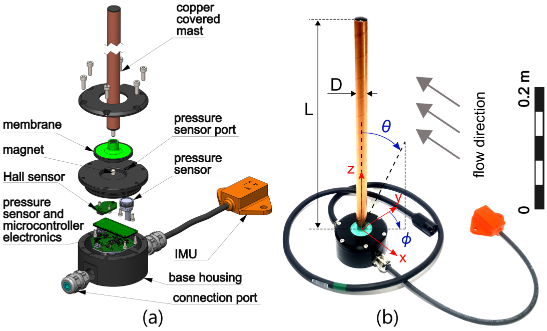


Fig. 1: Hydromast overview: (a) Hydromast exploded view; (b) Assembled Hydromast with axis x, y and z, mast tilt angle θ and direction ϕ, mast length L and diameter D.

The device consists of a CNC machined polyoxymethylene (POM) base, a flexible membrane and the mast. The mast is a hollow polycarbonate (PC) tube, closed at both ends and mounted to a flexible silicon membrane (Elite Double 22, Zhermak SpA). The mast has air inside that makes it positively buoyant and a higher natural frequency f_N compared to the water filled mast. Air filled mast is more sensitive to the flow and is therefore the preferred setup, as described later. The mast is covered with a $0.06\,\mathrm{mm}$ thin layer of copper to minimize biofouling during longer deployment periods. The standard diameter of the mast used in this work is D = $15\,\mathrm{mm}$, but the dimensions can easily be altered for specific applications. The length of the mast L can be varied depending on the needed measurement range.

The base housing incorporates PCBs of the microcontroller (Adafruit Feather with Atmel ATSAMD21 Cortex M0 processor), power, serial communication and pressure sensor. An absolute pressure sensor records the water height (86-030A-R, TE Connectivity) with the pressure port integrated into the POM casing (with range of 0-2 bar). The Hydromast can be connected directly over RS485 serial connection to PC or to a communication module with a raw data sampling rate of $50\,\mathrm{Hz}$. The Hydromast power consumption is approximately $0.15\,\mathrm{W}$ (at $5\,\mathrm{V}$ supply voltage). The cost of the Hydromast components is about 500€ at the time of writing this paper. The design of the Hydromast is shown in Fig.1a.

B. Working Principle

The Hydromast measures the mast location in x, y and z direction using magnetic field, with Hall sensor outputs X, Y and Z (in mT) accordingly. The coordinate system of the mast is shown in Fig. 1b. The magnitude

$$M_{XYZ} = \sqrt{X^2 + Y^2 + Z^2}$$

reading is observed to be linearly correlated to the tilt angle θ of the device, whereas the components X and Y correlate to the mast location in the $x - y$ plane and indicate the direction of the flow, noted as $\bar{\phi}$. Two different methods of flow velocity estimation can be introduced - (1) velocity estimate V_f based on the dominant frequencies f_d of the mast vibration (time-averaged estimate); (2) velocity estimate V_θ based on the tilt θ of the mast (instantaneous estimate).

Fluid-body forces govern the interactions between the mast due to VIV. Each Hydromast has it's own natural frequency f_N, that is the frequency the mast oscillates with, when disturbed and no forcing (i.e. flow) is present. When a cylindrical rod is put in a cross-flow, it generates vortexes with a vortex shedding frequency f_0. First velocity estimation method of the Hydromast is based on the vibrations of the elastically supported rigid mast as a VIV resonator. In [24], the design of the device was tuned so that the lightly damped cylinder would oscillate with frequency f as close as possible to the vortex shedding frequency f_0. It was shown that the time-averaged velocity can be estimated using the mean frequency spectral amplitude after taking the Fast Fourier Transform (FFT) of the mast vibrations (for more details refer to [26]). When the cylinder vortex shedding frequency is close to the natural frequency $f_0 \approx f_N$, an important lock-in phenomenon occurs: $f_0/f_N = 1$. In such cases, the shedding becomes controlled by the natural frequency, even if small fluctuations in the flow velocity occur. This and other resonance points of a lightly damped cylinder are related to the bulk flow by reduced velocity $V_r = V/(f_N D)$ where V is the velocity of the bulk flow and D the diameter of the cylinder [27]. For frequency based velocity estimate V_f, a relation based on the Strouhal number can be introduced, $St = f_0 D/V$. For the Hydromast, the relation between dominant frequency $f_d(M_{XYZ})$ and flow speed can be written as

$$V_f = \frac{D}{St} f_d + C_0, \tag{1}$$

where C_0 is a constant taking into account the end-effects (of the mast tip) and other artifacts of the specific device setup. For a stationary smooth circular cylinder for Reynolds numbers ranging from 10^3 to 10^5, the Strouhal number of vortex shedding is $St \approx 0.2$ [28]. Using $St = 0.2$ reduces Equation (1) to $V_f = \frac{D}{0.2} f_d + C_0$, where only C_0 needs to be empirically determined.

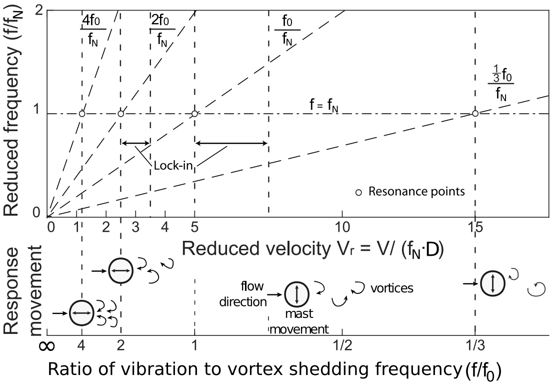


Fig. 2: Schematic response of lightly damped cylinder in crossflow (adapted from [27]).

In [26], a 100 mm long neutrally buoyant mast was used, which provided a working range from $0.5\,\mathrm{m/s}$ to $1.4\,\mathrm{m/s}$. One of the goals of this study was to extend the working range of the Hydromast, especially decrease the lower limit. This is done by choosing the mast with a correct natural frequency, based on the needed measurement range. Response to the cross-flow of a lightly damped circular cylinder is thoroughly described in [27]. If a cylinder is free to vibrate in any direction perpendicular to its axis, many modes of vortex shedding can occur. The most important excitation occurs from resonance when vibration frequency f becomes equal to natural vortex frequency f_N, where the vibration amplitude drastically increases and lock-in at the high amplitude vibrations is observed. This response mode starts around $f/f_N \approx 1$ and can at most last down to $f/f_N = 1/3$, in the range of reduced velocity of $5 < V_r < 15$, as schematically shown in Fig.2. This is also the range where the Hydromast vibrations occur and flow velocity can thus be determined, defining the measurement range for the Hydromast. Minimum and maximum velocities are found as

$$V_f min = 5 D f_N, \tag{2}$$

and

$$V_f max = 15 D f_N, \tag{3}$$

where f_N is the natural frequency in water.

For the second velocity estimate, V_θ, the exact position of the mast in time is determined using the Hall effect sensor outputs. This allows measuring tilt angle θ, nearly instantaneous flow speed V_θ as a function of θ, and flow direction $\bar{\phi}$. The tilt angle is assumed to be in linear correlation with the Hall sensor measurements, and can be calculated as $\theta = p_1 M_{XYZ} + p_2$, where p_1 and p_2 are empirically detected calibration constants.

Mast tilt angle θ is linearly dependent on the magnitude of the magnetic field and therefore velocity can simply be correlated to M_{XYZ} as

$$V_\theta = C_1 M_{XYZ} + C_2, \tag{4}$$

where constants C_1 and C_2 are found through calibration. Magnitude itself is used here, instead of θ in order to avoid introducing unnecessary extra uncertainty, as θ itself is also determined using calibration fit. It is important to note that this instantaneous velocity measurement based on mast tilt V_θ is independent from the frequency based measurement V_f and can therefore be used as separate measurement quantity.

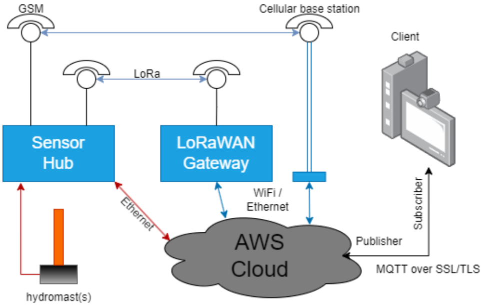


Fig. 3: Communication framework overview.

The flow direction $\bar{\phi}$ can be found directly using Hall effect sensor outputs X and Y as instantaneous mast direction $\phi = \text{atan2}(X, -Y)$ which represents the mast location. To find the flow direction, an average over several vibrations must be calculated, taking the median value over some time, denoted with $\bar{\phi}$.

Additional to flow measurements, the Hydromast can also be used to measure the water level changes. The device is equipped with a pressure sensor enabling continuous measurement of the water level above.

C. Communications

In addition to modifying the sensing methods and hardware, the Hydromast now incorporates communication capabilities, enabling near real-time flow monitoring and device failure detection [30], [31]. These enhancements have expanded the range of potential applications for the device. To achieve this, a sensor hub based on Raspberry Pi3 is utilized (we have also tested Beaglebone, and it is possible to use another ARM-based single-board computers). Fig.3 shows the overview of the communication framework. The sensor hub stores data locally and transmits it to Amazon Cloud IoT Core using the MQTT protocol. In addition, the sensor hub can receive configuration updates from MQTT, specifically the connected mast types and calibration constants (see III). Infrastructure as code (IaC), specifically AWS Cloud Development Kit (CDK), is employed for configuring and managing AWS services, which allows consistent configurations and stronger security. The source code for both self-developed components and IaC is maintained in GIT repositories. Python 3.9 and ReactJS are used for programming the self-developed components, including the web interface.

The sensor hubs operate on self-developed application in Docker container running on BalenaOS, and their management is facilitated through BalenaCloud servers. For communication with the Amazon Cloud, various options are available, including the use of GSM network, LoRaWAN radio connection, and Ethernet communication. The proposed networking framework makes the Hydromast deployment both secure and scalable but also efficiently manageable from the sensors fleet point of view.

III. CALIBRATION AND CHARACTERISATION

In order to demonstrate the behavior and characteristics of the device, characterisation was first performed in lab, after which the field tests were done for calibrating the device.

A. Natural frequencies and measurement range

For simple measurement range estimation, the mast's natural frequency dependence on the length was characterized. Natural frequencies in water and in air for masts with varying length and mass were measured. The relation between L and f_N in water was represented by a power fit $f_N \sim L^a$, where a is a constant, whereas the ratio of natural frequencies was found to be $f_{Nair}/f_{Nwater} \approx 1.4$. The natural frequencies and corresponding sensor measurement limits $V_f min$ and $V_f max$ are shown in Table I. Using these estimated ranges, the mast length can be chosen based on the measurement range needed in each specific application. For an extended measurement range, a setup with multiple Hydromasts can also be used, so that depending on the flow velocity, data from correct device is acquired. It was chosen to continue with a positively buoyant mast due to it's higher natural frequencies and faster response to the flow changes, to better capture unsteady flow phenomena. It must be noted that the tension restraint force to counteract buoyancy is larger for longer masts (due to larger volume) which can also change the performance characteristics.

For validation purposes, two masts were chosen: HM300 with L=300 mm and HM200 with L=200 mm length, both highlighted in Table I. These two masts together cover a range from V= 0.1 to 0.7 m/s being sufficient for many applications and they also have a wide enough range overlap to make comparisons.

B. Tilt angle calibration

Tilt angle was expected to be linearly correlated to the Hall sensor's output magnitude M_{XYZ}. A table-top calibration was performed, in order to determine the calibration coefficients. The mast was tilted at 11 different angles from $0 < \theta < 30\,\text{deg}$ in 8 different directions with 45 degree increments. The mast tilt was recorded using a Go-Pro camera and the tilt angle was calculated from the image. The angle showed a very good linear correlation with the output magnitude M_{XYZ}, as shown in Fig.4. The calibration constants were found to be p_1=1.2 and p_2=1.13, with a coefficient of determination R^2 of 0.98.

C. In-flow Velocity Calibration

The Hydromast velocity calibrations need to be determined in real flow conditions, where the turbulence levels, velocity profile and set-up are similar to the planned field applications. The Hydromasts were first tested in a small lab-scale flow-channel. However, as the channel was too small for these devices and flow conditions were significantly different from real flow, this proved to be too unreliable and inaccurate for calibration purposes. Therefore, all calibrations characterization were performed in a natural river flow.

TABLE I: Characteristics of Hydromasts based on mast length

Mast	L [mm]	mass m [g]	Buoyancy	f_N water [Hz]	f_N air [Hz]	$V_f min$ [m/s]	$V_f max$ [m/s]
HM50	50	9.4	positive	17.3	25.0	1.30	3.90
HM100	100	13.1	positive	7.5	12.2	0.56	1.69
HM100NB[1]	100	20.6	neutral	6.9	9.1	0.52	1.55
HM150	150	21.9	positive	4.9	6.5	0.37	1.10
HM150NB[1]	150	30.3	neutral	3.7	5.0	0.27	0.82
Gray HM200	200	27.5	positive	3.1	4.2	0.23	0.69
Gray HM300	300	39.4	positive	1.7	2.2	0.13	0.39

[1] NB - neutrally buoyant mast

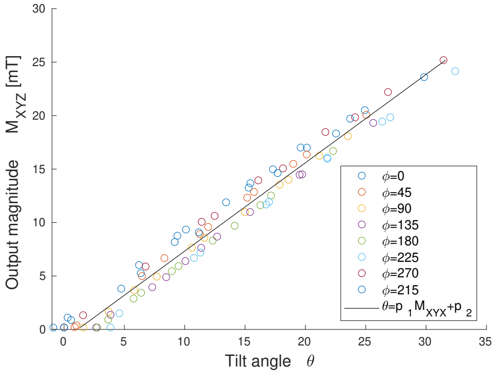
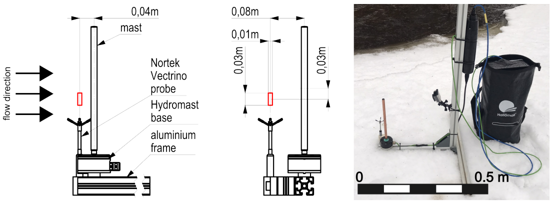


Fig. 4: Hydromast tilt angle calibration. Output magnitude M_{XYZ} for varying tilt angle θ with the linear fit. Measurements shown for 8 different directions from $\phi = 0$ to $\phi = 315$.

The calibrations for flow velocity were conducted in Keila river (latitude 59.394537 N, longitude 24.294915 E; closest address Posti 1, Keila-Joa, 76701 Harjumaa, Estonia) with average water level of 102 cm measured at Keila river hydrological station [32]. The aim was to calibrate the Hydromast against ADV measurements (Vectrino Profiler, Nortek, Norway) in real-world conditions. ADV was chosen as reference because it is commercially available standard device for this type of flow measurements. An overview of the calibration setup is shown in Fig.5. One Hydromast was deployed together with an ADV for reference measurements. Samples with a duration of 60 seconds were taken at 47 locations for HM300 and 29 locations for HM200. The Hydromast and ADV measurements were synchronized by using the sync signal from the ADV to trigger the recording of the Hydromast data logger (based on a Raspberry Pi 3, Raspberry Pi Foundation in association with Broadcom).

For all calibration and steady flow experiments, the Hydromast data acquisition rate was 50 Hz. When calculating V_f, no pre-processing of the data was done, the average velocity estimate was based on the frequency spectrum of the raw data over the duration of the sample. For calibration, 60 s long samples were first recorded and by looking at the convergence it was determined that at least 20 s sample is needed for an accurate dominant frequency estimate. Based on this, the calibration and validation sample length was chosen to be 30 s. Similarly, the tilt based velocity estimates, V_θ, were first done based on raw data. As the interest lied in the time-averaged velocities, and average V_θ was calculated over the duration of the sample using $\tilde{V_\theta} = median(V_\theta)$.

In the calibration as well as steady flow validation experiments, the ADV data acquisition rate was 25 Hz and the ADV data processing was kept minimal. All steady flow experiments were done in real conditions and close to the surface. Hence, to reduce the noise in the averages due different natural and device caused phenomenon (more details about ADV noise sources can be found in [29]), the median ADV velocity was also calculated over the duration of the 30 s sample.

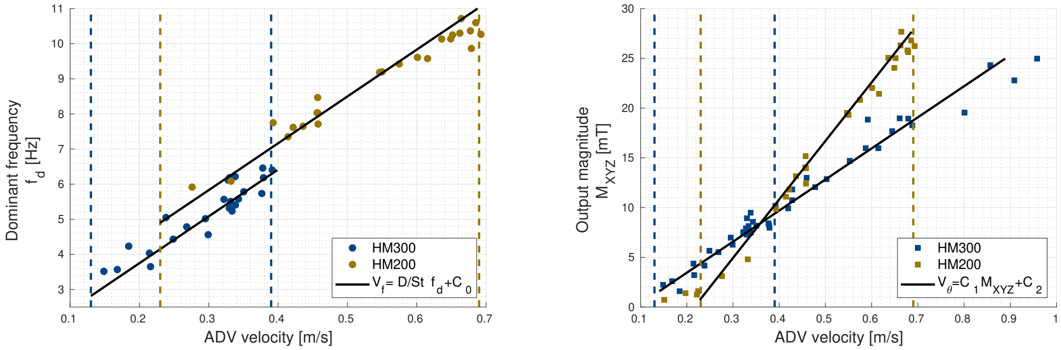


(a) (b) (c)

Fig. 5: Setup for the in-flow velocity calibration in a river. (a) Side view and (b) front view with the ADV and Hydromast (HM300) placement; (c) assembled setup in the field.

Fig.6a shows the calibration data for HM300 and HM200 together with the calibration curves for the dominant frequency based velocity estimate V_f. It can be seen, that a linear fit with constant slope D/St and calibration constant C_0 is a good approximation and describes well the relation between flow velocity and mast vibration, with $R^2 = 0.87$ for HM300 and $R^2 = 0.96$ for HM200. For calibration, all data was used, where a clear energy peak in the frequency spectrum could be determined. For HM300, peaks were detected for velocities from 0.15 to 0.50 m/s and for HM200 the range was from 0.27 to 0.70 m/s. These ranges agree well with the theoretical estimates from Table I, according to which HM300 should work from 0.13 to 0.39 m/s and HM200 from 0.23 to 0.69 m/s. These calibrations results agree well with the analysis described in Section II-B and support the theoretical assumptions. This calibration shows that dominant frequency can be used for flow velocity estimation and also, if needed, several devices with different ranges can be used for extending measurement range.

Another method of determining velocity is using the tilt angle θ. Calibration data and linear fits for this method with both Hydromasts are shown in Fig.6b. Here, all calibration points are shown (averaged over 30 seconds) and it can be seen



(a) (b)

Fig. 6: Velocity calibrations of 200mm and 300mm masts: (a) Dominant frequency based velocity V_f calibration. (b) Tilt based velocity V_θ calibration. Calibration fit shown as solid line, theoretical velocity limits $V_f min$ and $V_f max$ are shown as dotted line.

that both HM300 and HM200 have a linear dependence on the magnitude M_{XYZ}, with R^2 =0.96 for HM300 and R^2 =0.96 for HM200. The lowest speeds, at which the velocity can be measured, are 0.15 and 0.22 m/s for HM300 and HM200 respectively, being similar to the lowest limit where V_f can be used. HM200 has a higher slope, making it more sensitive to velocity changes, whereas HM300 has a slightly lower sensitivity. The tension restraint force to counteract buoyancy is almost double for HM300 mast compared to HM200 and this seems to make the longer mast HM300 less sensitive to tilt due to flow velocity. This lower sensitivity constitutes a close to a five times bigger measurement range for the HM300, with new range spanning from 0.15 m/s up to nearly 1 m/s. This is a large increase in range of the velocity measurement for HM300 compared to the working range for V_f (which was up to 0.39 m/s), expanding the applicability of the Hydromast significantly. Hence, the tilt provides an accurate estimate for flow velocity with extra benefit of having a wider measurement range compared to the frequency based estimate for HM300. It must be noted, that the membrane of each Hydromast is currently hand-made and therefore the tilt calibration can also be slightly affected by the membrane properties. Therefore sometimes it can be useful to check with V_f the validity of the V_θ calibration, if devices with new membranes are used.

In order to have independent measurements with the Hydromast, it is important for the device to detect when the measurements are in the working range, and recognize and remove outputs when the Hydromast is out of range providing unreliable data. For V_f this 'out of range' criterion is defined by setting a minimum limit for spectral amplitude level at dominant frequency, so that only distinct high energy peaks are detected. For the Hydromast analysis, two dominant frequency peaks are detected in the spectrum (due to the 8-shaped movement of the mast) and the second one, at higher frequency, representing the cross-flow vibrations, is chosen to be the estimate. This criterion depends on the length of the Hydromast and is determined based on the calibration data. For tilt angle based estimate V_θ the 'out of range' criterion is simply the minimum and maximum values for the magnitude M_{XYZ}, also determined based on the calibration results.

IV. VALIDATION

The validation of the Hydromast velocity estimations was performed in three stages. First, validation was performed in steady flow conditions in a river, with many 30 s measurements. In the second stage, two devices were installed for a long period in steady flow in a river, to show long-term data and validate communication capabilities. This flow is steady in short-term but has variations over longer time. Finally, the validation was performed in unsteady flow conditions on the coastline to demonstrate and validate the capability of the Hydromast measuring unsteady flows.

A. Steady flow, short-term tests

The short-term validation tests were performed on the same site as used for calibration, but at different flow conditions. Measurements were taken in many locations in the river to show the performance of the device with the same setup at many different flow speeds. Measurement points were chosen taking into account the working ranges of the Hydromasts and steady flow conditions.

A setup with two Hydromasts, HM300 and HM200, together with the ADV (Vectrino Profiler) was assembled as shown in Fig.7. The validation was performed in Keila river on the same location as the calibration experiments described in section III, with a higher water level (water level of 112 cm measured at Keila water level station [32]) which allowed finding locations with a wide range of flow velocities. A total of 139 samples were collected over a range of velocities from 0.01 to 1.05 m/s.

The results of the velocity estimations based on the mast vibrations are shown in Fig.8a and Fig.8b for HM300 and HM200 respectively. ADV measurement serves as our reference velocity. It must be noted that also these measurements have uncertainty and to visualize that, the ADV measurements

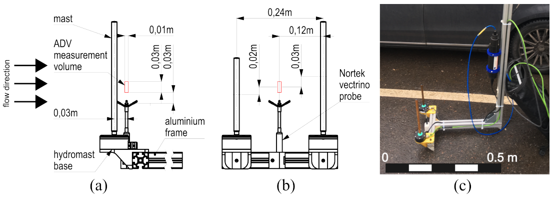


Fig. 7: Setup for steady flow validation tests in a river. (a) Side view and (b) front view with the ADV and Hydromast (HM300 and HM200) placement; (c) assembled setup in the field.

are shown with standard deviation (SD) (shaded in grey), which can be considered as the uncertainty of the reference measurement. Points, which according to ADV have high turbulence intensity (TI) levels, above 40%, are indicated with pink in the figures.

Comparison of the V_f from the longer HM300 with ADV is shown in Fig.8a. Hydromast data agrees well with the reference ADV, having root-mean-square-error (RMSE) of $0.065\,\mathrm{m/s}$ and all velocities detected lay within the estimated measurement range, between $V_f min$ and $V_f max$, indicated as a shaded box in the figure. In this case the majority of the data detected lies within the theoretical measurement range, supporting the theoretical model. The used criteria of minimum energy level at dominant frequency works well for peak detection, only showing data that is within the theoretical range. Turbulent measurements have a slightly higher variation but overall there is also a good agreement to the reference measurement, showing that no significant change in data quality is introduced.

In Fig.8b the Hydromast HM200 data is shown for V_f. The measurements agree well with the ADV within the estimated measurement range. Above 0.6 m/s dominant frequencies are still detected but seem to drift away from the reference value. This change is probably due to a mode shift in vibration occurring near $V_r = 15$. It was seen that above this value also tilt does not change anymore, the vibration is somewhat altered and does not represent velocity changes well, even though peaks in the energy spectrum are still present. As there was no good indicator found in the spectra to filter the 'out of range' data, V_θ could be used as an indicator of the velocity range, so that if V_θ is out of its measurement range, no output of V_f is provided. Points with this criteria applied are indicated in Fig.8b as empty symbols. For all the data, the RMSE value is $0.113\,\mathrm{m/s}$, whereas the data with V_θ based criteria (filled symbols) has RMSE $0.03\,\mathrm{m/s}$. Alternatively, an empirical calibration fit could be used on the data, which follows the higher end frequencies better and accounts for that mode change. But with this approach there is the downside that each membrane needs an extended calibration in similar flow conditions as the application.

Tilt based velocities from HM300 are shown in Fig.9a. Below $0.15\,\mathrm{m/s}$, which is the lower theoretical measurement limit, the measurements have a constant value and are not usable (empty symbols indicate 'out of range' data). There is an agreement between ADV and Hydromast from $0.15\,\mathrm{m/s}$ up to even $1.0\,\mathrm{m/s}$, with RMSE of $0.047\,\mathrm{m/s}$ excluding turbulent and out of range points (RMSE is $0.093\,\mathrm{m/s}$ including all points). The measurement range is higher than the one achieved in calibration, but seems that the lower sensitivity to flow velocity (as discussed above in section III) allows much wider measurement range. This demonstrates that the HM300 can be used on its own for a wide range of velocity measurements from 0.1 to $1.0\,\mathrm{m/s}$, compared to V_f measurement range being only one third of it. Only some of the high turbulence data at higher velocities is not following the trend, therefore care needs to be taken at very turbulent conditions at high flow speeds (higher than V_f range).

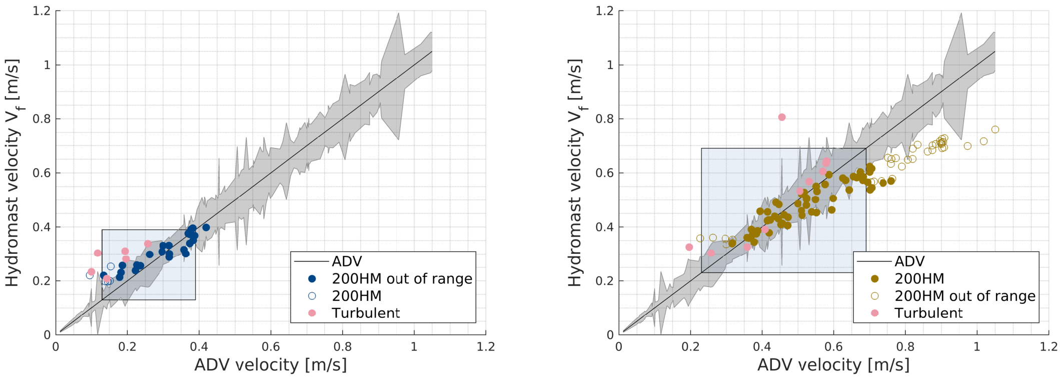
Fig.9b shows results for tilt based V_θ for HM200. At low velocities the 'out of range' data is again constant as for the other mast (hollow symbols). Above that there is an excelent agreement between ADV and Hydromast measurements between $0.3\,\mathrm{m/s}$ and $0.7\,\mathrm{m/s}$, where in-range data is marked with filled symbols and having RMSE of $0.03\,\mathrm{m/s}$. Above that the HM200 has reached its maximum tilt angle and does not capture flow velocities above $0.7\,\mathrm{m/s}$. Out of range data based on magnitude limits is shown with empty symbols and it can be seen that this criterion works well for these velocities, filtering out data which does not represent correct velocities. For HM200, the working range of V_θ is the same as the theoretical range of V_f in Table I.

Overall, both velocity estimates agree well with the reference measurements. Some of the high turbulent data do not follow general trends, but there seems not to be any systematic impact. In high turbulence conditions and high speeds the data has lower accuracy and could overestimate the flow speed, whereas at lower speeds high turbulence intensity does not seem to affect the results.

A set of measurements were also performed to evaluate the flow direction. For direction, no calibration is needed, as the direction $\bar{\phi}$ can be calculated directly from the Hall sensor output. For this, the full experimental setup was rotated by 45 degree increments and measurements from Hydromast and ADV were compared. This was done with both devices, at several flow speeds, to test the sensitivity at slow, medium and high velocities. The measured angles compared with the ADV reference are shown in Fig.10. Both Hydromasts show a very good direction estimate, having RMSE of 3.46°. The variaton is bigger in places where also ADV had higher RMS, showing that there was higher variability in flow, resulting in less reliable results.

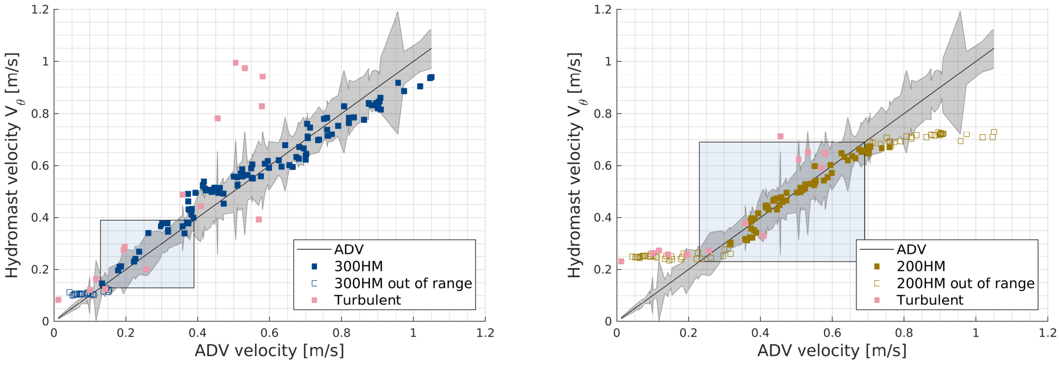
B. Steady flow, long-term tests

A long-term river test was performed in order to verify the Hydromast durability and performance, as well as demonstrate the near real-time flow monitoring. Two devices were deployed, in the same configuration as shown in Fig.7. Validation measurements were taken with an ADV (Vectrino Profiler, Nortek AS, Norway) every second day over a two week testing period. The validation test was carried out in Vääna river (latitude:59.292888 N, longitude 24.739287 E; closest address Otto tee, Lokuti, 75514 Harjumaa, Estonia).



(a) (b)

Fig. 8: Steady flow short-term validation data for frequency based velocity estimate V_f.(a) 300mm mast; (b) 200mm mast. Filled symbols denote the data in range, empty symbols out of range. SD ranges from ADV measurements are shaded in grey. The blue box indicates the theoretical measurement range.



(a) (b)

Fig. 9: Steady flow short-term validation data for tilt based velocity estimate V_θ .(a) 300mm mast; (b) 200mm mast. Filled symbols denote the data in range, empty symbols out of range. SD ranges from ADV measurements are shaded in grey. The blue box indicates the theoretical measurement range.

In this experimental setup, the devices were streaming data online with the framework described in section II-C. The 2 Hydromasts were connected to a Raspberry Pi 3 (Raspberry Pi Foundation in association with Broadcom) microcomputer running Balena OS, that was battery and solar panel powered over the whole testing period. The live data stream monitoring over GSM network allowed to evaluate the Hydromast performance as well as detect faults in the measurements. This allowed near real-time monitoring (about 3s latency) of river flow velocity, water level and also direction.

The average velocities throughout the two-week tests for both HM200 and HM300 are shown in Fig.11a. The frequency based velocity V_f was estimated for 30s intervals and averaged over 30 minute periods. The tilt based V_θ has been calculated as instantaneous velocity and averaged over 30 minute periods. These tests were ran during spring entering into the dry season, hence, the river water flow velocity decreases in time. Both V_θ and V_f show a steady decrease in flow velocity from 0.37 to 0.2 m/s and agree well with the ADV reference measurements taken. For tilt based V_θ from HM300, higher variations in speed are captured compared to V_f. For higher speeds during first days V_θ seems to overestimate compared to the reference velocities, but both measurements are within the ADV standard deviation range. After 4th of April, where speeds are lower, the agreement with ADV reference measurements becomes very good for both of the velocity estimates.

As for HM200, V_θ and V_f measure continuously during the first days and agree very well with each other and with reference ADV measurements. After 4th April 2023 flow velocity started to fall below the measurement range of the HM200 and it does not give an accurate estimate anymore. For V_θ, M_{XYZ} goes below minimum limit defined earlier (data marked as light grey) and for V_f, fewer high energy peaks are detected.

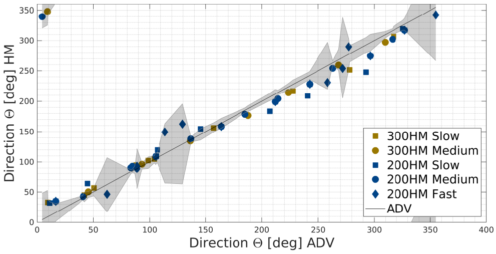


Fig. 10: Flow direction validation data. The direction measurements were done in varying flow conditions slow (up to $0.22\,\mathrm{m/s}$), medium (up to $0.61\,\mathrm{m/s}$) and fast (up to $0.85\,\mathrm{m/s}$)).

In addition, flow direction was estimated for the same experiments, and the results are shown together with the ADV direction in Fig.11b. As there was no specific device orientation reference taken at the test site, comparison with ADV was made using the first measurement point. Based on that, a constant offset of 5° was removed from the Hydromast data.

In Fig.11c the depth estimates using the Hydromast are shown. Tallinn-Harku weather station data [33] was used as the atmospheric pressure reference and manual measurements were taken for comparison during ADV measurements. Both Hydromasts behave similarly well and agree with the measurement points, all estimates varying within 10 cm range. HM200 does not follow the trend during the last two days and this is due to a failure of the pressure sensor, which was determined after the tests.

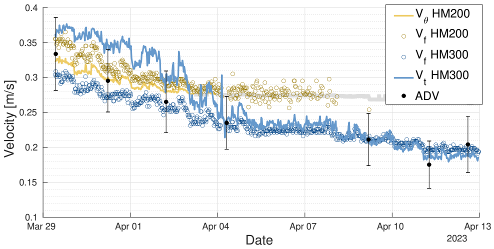
C. Unsteady flow

Several tests were conducted in the sea, near the coast, to validate the behavior of the Hydromast in unsteady waves with varying flow direction and magnitude. Measurements were done for 10 minute time periods, in order to allow long enough data sets for spectral analysis with acquisition rate of 50 Hz. A commercial ADV (Vector, Nortek AS, Norway) was used as the reference measurement device, with acquisition rate of 8 Hz. Two Hydromasts, HM200 and HM300, together with the ADV probe in between were installed on a solid frame and immersed in sea on a sandy flat surface, at about 0.9 m depth. The experimental setup of this test is shown in Fig.12.

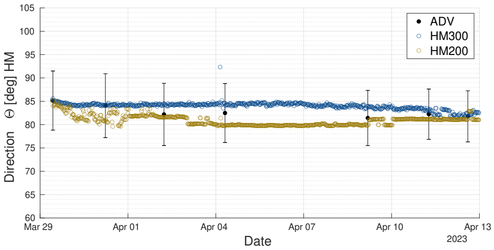
To validate the unsteady velocity estimations, measurements were taken simultaneously for the two Hydromasts and ADV. The experiments were done on two days: on 4.May 2023 in Pikakari beach (lat:59°28'26.2"N long:24°43'27.4"E) and on 18. May 2023 at Vääna beach (lat:59°25'29.0"N long: 24°20'19.1"E). Locations and days were chosen to test devices at different flow conditions. For both, the Hydromast and the ADV, the first velocity estimations were done on raw data. The resulting velocity estimations were then passed through a lowpass filter with cutoff frequency $2\,\mathrm{Hz}$ for consistency and better comparisson.

On the 4th of May, the conditions were very calm, the maximum velocity reached was 0.3 m/s. These velocities were below HM200 measurement range, therefore only HM300 data is analysed and shown. Figure 13 shows 5 minutes of the measurements of ADV and HM300. It can be seen that the Hydromast shows similar behavior in velocity magnitude as the reference ADV. It can clearly be seen that HM300 follows nicely the same trend as the reference measurement for higher flow speeds, demonstrating the capability of the Hydromast to estimate instantaneous flow velocities in unsteady flow.

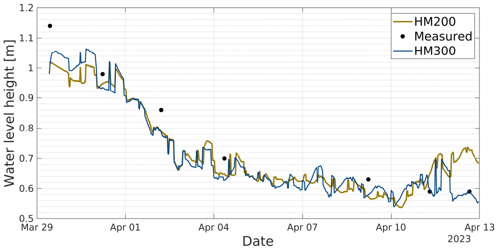
The unsteady measurements on 18.May 2023 are shown in figures 14a and 14b, for HM300 and HM200, respectively. In this case the velocity was within the measurement range for both devices. HM300 shows a very good correlation throughout the data, following all the ADV peaks closely, especially well seen at the zoom-in. HM200 in Fig.14b shows agreement with ADV at higher speeds but is cutting off the lower velocities, as it is not sensitive enough at low speeds. At speeds above the minimum theoretical range a very good correlation with the reference measurement can be



(a)

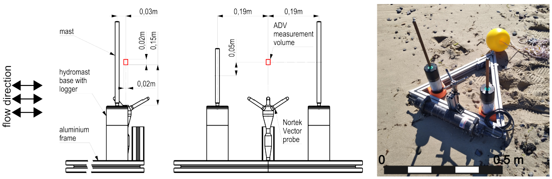


(b)



(c)

Fig. 11: Steady flow long-term validation data in Vääna river. Results averaged over 30 minutes. (a) Frequency and tilt based velocity estimates; (b) Flow direction ϕ; (c) Water level height.



(a) (b) (c)

Fig. 12: Setup for unsteady flow validation tests in the sea (a) Side view and (b) front view with the ADV and Hydromast placement; (c) assembled setup in the field.

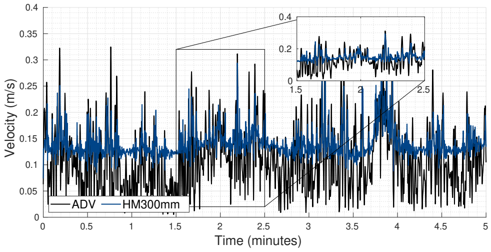


Fig. 13: Unsteady flow tilt based velocity validation test Pikakari beach, 4.May 2023, 300mm mast.

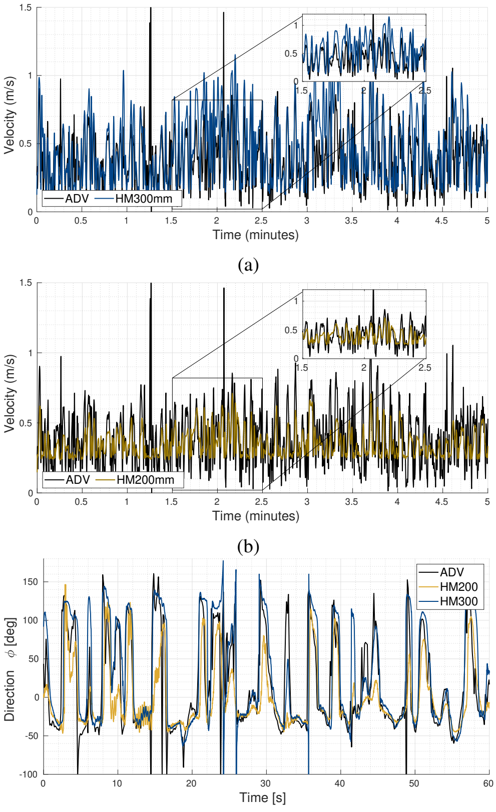
observed. The root-mean-square deviation from ADV data was $0.095\,\mathrm{m/s}$ for HM200 and $0.101\,\mathrm{m/s}$ for HM300.

In addition, both HM200 and HM300 were used to estimate the flow directions during the 18th May experiments when the flow speed was high enough to work with both masts. A short 1 minute segment of the directions is shown in Figure 14c. The direction varies a lot as the flow near coast have short waves due to wind and swell. Direction estimations from both devices show very good agreement when compared to the ADV, having equally fast reaction to the change of the direction.

V. DISCUSSION

The characterization and validation of the Hydromast with the Hall effect sensor has been performed. Both velocity estimates, the frequency based velocity V_f and the tilt based V_θ, perform well in measuring flow velocities in various setups. For V_f, the functional dependency as well as the working range agree well with the described theoretical framework. Further, the longer mast HM300 resolves well lower velocities and can be easily used as an independent measurement. For the shorter mast HM200, the velocity range is wider, but an extra range criteria based on V_θ needs to be used to detect the measurement range and allow accurate velocity estimates. However, HM200 measurements are more robust, whereas overlap for HM300 is relatively small for V_f and V_θ.

When calculating average flow speeds over longer periods of time using HM200, care needs to be taken interpreting the low velocity data. Around the lower limit of the measurement range frequency based estimations only occur for higher velocities for HM200 and not for lower, which can lead to biased results. One option would be using estimate from tilt as an indicator, if V_f is reliable, similar to what was suggested for upper velocities in the short validation tests. Further, a minimum number of samples required for the average can be implemented.



(a)

(b)

(c)

Fig. 14: Unsteady flow validation tests, Vääna beach, 18.May 2023. (a) tilt based velocity, 300mm mast; (b) tilt based velocity, 200mm mast; (c) direction ϕ, 200mm and 300mm mast.

As for the tilt based V_θ, for HM300, the measurement range is three times larger than for V_f, allowing the device to be used in applications with high velocity variations. For HM200, range is comparable with V_f, giving independent and reliable velocity estimates. Using V_θ, instantaneous velocities in changing flow conditions can be measured, allowing measurements in areas where flow direction is constantly changing, like with the waves on the coast. Here again, HM300 seems to perform better, capturing lower velocities than HM200 and both sensors seem to capture higher velocities.

In unsteady flow, the ADV velocity and the Hydromast velocity showed good correlation. However, the measurements were taken in not ideal conditions for the ADV, namely the experiments were done in shallow water (i.e., ADV was both close to the bottom and near the surface). For more robust unsteady flow characterisation, additional work is needed to

TABLE II: Uncertainty estimates (95% confidence level).

Mast	V_f [m/s]	V_θ [m/s]	$\bar{\phi}$ [deg]	Water level [m]
HM200	0.096	0.050	12.10	0.004
HM300	0.049	0.094	12.02	0.003
ADV	0.129	0.129	36.45	N/A

accurately describe the Hydromast reaction time for instantaneous measurements.

Direction estimate and water column height measurements were done against reference measurements in different flows. The estimates agreed between devices as well as with the references, showing that they are reliable extra measurements that can be taken during testing. In steady river flow there was very little variation and the results just show the stability of the direction measurement in time, which agrees with ADV estimates within 5 degrees. The Hydromast has the capability of a near real-time data monitoring, when a sensor hub can be mounted above water. This is useful for long-term measurements and is also helpful for fault detection.

Based on the validation data, the overall uncertainties of the measurements were estimated for the 95% confidence level, reported in Table II. Same estimates for ADV have also been shown for comparison. As both velocity estimates have their own pros and cons, it could be considered to use the two independent velocity estimates in parallel and combine them for higher accuracy. Alternatively, for higher accuracy, calibrations can be performed before the actual tests in similar flow conditions to capture the specific behavior. High turbulence levels would provide rapid changes in velocity output and that could be an indicator that caution needs to be taken in data interpretation.

The Hydromast measurements are independent of the water quality and surface reflections, allowing it to be used also in locations, where acoustic methods fail. In future, more reliable pressure sensors should be used in order to avoid drifting and providing more reliable water depth data. Additional research could be done on fault detection and unwanted debris detection. Also, finding a way to estimate turbulence intensity from the Hydromast output would be a useful feature, in order to indicate high turbulence conditions.

The compact design and affordable price of the Hydromast (roughly $1/10^{th}$ of a commercial ADV) coupled with its versatile communication capabilities, make it suitable for a wide range of applications in shallow water environments. Its low cost enables distributed sensing in various applications, including improving safety in harbors by monitoring currents, detecting ship traffic along coastlines, and evaluating bed load for sediment transportation studies. Furthermore, the distributed sensing capability allows for the application of these devices in the aquaculture industry, as well as in the field of renewable energy, for site monitoring and site evaluation.

VI. CONCLUSIONS

In this paper the Hall effect sensor based low-cost flow monitoring device Hydromast has been introduced. The device was characterized and validated against a commercial acoustic Doppler velocimeter and shown to perform well in various flows, with flow speeds from 0.15 to 1 m/s. With this new device, average and instantaneous flow speed along with flow direction and water depth can be measured, allowing the devices to be used in both steady and fluctuating flow conditions. Cloud communication functionalities were developed so that monitoring can be done online, allowing long term testing with live outputs and data analysis, as well as allowing fault detection. With low cost and high reliability in near-bed flow velocity estimations, the described device can be used in various flow conditions for flow velocity and direction estimation, both short as well as long-term flow monitoring, as a single device or in a larger grid with a near real-time data output to the user.

VII. ACKNOWLEDGMENT

The authors would like to thank Jaan Rebane and Andres Ernits for their support with Hydromast manufacturing and Prof. Jeffrey Andrew Tuhtan for the fruitful discussions.

REFERENCES

[1] Dingman, S. Physical hydrology. (Waveland press,2015)

[2] Kalmijn, Ad J. Hydrodynamic and acoustic field detection. *Sensory biology of aquatic animals*. New York, NY: Springer New York, pp. 83-130.(1988)

[3] Betteridge, K., Williams, J., Thorne, P. & Bell, P. Acoustic instrumentation for measuring near-bed sediment processes and hydrodynamics. *Journal Of Experimental Marine Biology And Ecology*. **285** pp. 105-118 (2003)

[4] Kostaschuk, R., Best, J., Villard, P., Peakall, J. & Franklin, M. Measuring flow velocity and sediment transport with an acoustic Doppler current profiler. *Geomorphology*. **68**, pp. 25-37 (2005)

[5] Elfrink, B. & Baldock, T. Hydrodynamics and sediment transport in the swash zone: a review and perspectives. *Coastal Engineering*. **45**, pp. 149-167 (2002)

[6] Shields, F. & Rigby, J. River habitat quality from river velocities measured using acoustic Doppler current profiler. *Environmental Management*. **36** pp. 565-575 (2005)

[7] Levin, L., Boesch, D., Covich, A., Dahm, C., Erséus, C., Ewel, K., Kneib, R., Moldenke, A., Palmer, M., Snelgrove, P. & Others The function of marine critical transition zones and the importance of sediment biodiversity. *Ecosystems*. **4** pp. 430-451 (2001)

[8] Wu, Wenyan, et al. Ensemble flood forecasting: Current status and future opportunities. *Wiley Interdisciplinary Reviews: Water* 7.3 (2020): e1432.

[9] Recking, A. Theoretical development on the effects of changing flow hydraulics on incipient bed load motion. *Water Resources Research*. **45** (2009)

[10] Bockelmann, B., Fenrich, E., Lin, B. & Falconer, R. Development of an ecohydraulics model for stream and river restoration. *Ecological Engineering*. **22**, 227-235 (2004)

[11] Tomsett, Christopher, and Julian Leyland. Remote sensing of river corridors: A review of current trends and future directions. *River Research and Applications* 35.7 (2019): 779-803.

[12] Dolcetti, G., et al. "Using noncontact measurement of water surface dynamics to estimate river discharge." Water Resources Research 58.9 (2022): e2022WR032829.

[13] Planet inc, SkySat, Planet Developers, Available online: `https://developers.planet.com/docs/data/skysat/` (accessed on 17 July 2023)

[14] European space agency, SkySat, European space agency, Available online: `https://earth.esa.int/eogateway/missions/skysat#instruments-section` (accessed on 17 July 2023).

[15] Stone, M. C., & Hotchkiss, R. H. (2007). Evaluating velocity measurement techniques in shallow streams. Journal of Hydraulic Research, 45(6), 752-762.

[16] Plant, W., Keller, W. & Hayes, K. Measurement of river surface currents with coherent microwave systems. *IEEE Transactions On Geoscience And Remote Sensing*. **43**, 1242-1257 (2005)

[17] Mied, R., Chen, W., Smith, G., Wagner, E., Miller, W., Snow, C., Marmorino, G. & Rhea, W. Airborne remote sensing of surface velocities in a tidal river. *IEEE Transactions On Geoscience And Remote Sensing*. **56**, 4559-4567 (2018)

[18] Hauet, A., Creutin, J. & Belleudy, P. Sensitivity study of large-scale particle image velocimetry measurement of river discharge using numerical simulation. *Journal Of Hydrology*. **349**, 178-190 (2008)

[19] M. Marian, J. Kim, D. Kim, "Impact of the sampling duration on the uncertainty of averaged velocity measurements with acoustic instruments", *Hydrological processes*, vol. 35, no. 4, Apr. 2021.

[20] U. C. E. Zanke, "On the influence of turbulence on the initiation of sediment motion." International Journal of Sediment Research 18.1 (2003): 17-31.

[21] E. Muller, H. Decamps, and M. K. Dobson. "Contribution of space remote sensing to river studies." Freshwater Biology 29.2 (1993): 301-312.

[22] M. Rätsep, K. E. Parnell, T. Soomere, M. Kruusmaa, A. Ristolainen, J. A. Tuhtan. Surface vessel localization from wake measurements using an array of pressure sensors in the littoral zone. *Ocean Engineering*, 233, 109156, (2021

[23] P. Wang, X. Tian, T. Peng, and Y. Luo. "A review of the state-of-the-art developments in the field monitoring of offshore structures." Ocean Engineering 147 (2018): 148-164.

[24] A. Ristolainen, J. A. Tuhtan, A. Kuusik, and M. Kruusmaa, "Hydromast: A bioinspired flow sensor with accelerometers," *Biomimetic and Biohybrid Systems*, vol. 9793, N. F. Lepora, A. Mura, H. G. Krapp, P. F. M. J. Verschure, and T. J. Prescott, Eds. Berlin, Germany: Springer, pp. 510–517, 2016.

[25] A. Ristolainen, K. Kalev, J. A. Tuhtan, A. Kuusik, and M. Kruusmaa, "Hydromorphological classification using synchronous pressure and inertial sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, no. 11, pp. 1–11, Nov. 2018.

[26] A. Ristolainen, J. A. Tuhtan and M. Kruusmaa, "Continuous, Near-Bed Current Velocity Estimation Using Pressure and Inertial Sensing," *IEEE Sensors Journal* vol. 19, no. 24, pp. 12398–12406, Dec 2019.

[27] E. Naudascher and D. Rockwell, *Flow-Induced Vibrations: An Engineering Guide.* New York, NY, USA: Dover, 2005.

[28] F. M. White, *Viscous Fluid Flow.* 3rd ed. New York, NY, USA: McGraw-Hill, 2006.

[29] G. Voulgaris & J. H. Trowbridge, (1998). Evaluation of the acoustic Doppler velocimeter (ADV) for turbulence measurements. Journal of atmospheric and oceanic technology, 15(1), 272-289.

[30] L.Vihman, J.Raik, "Adaptive Kalman Filter Based Data Aggregation in Fault-Resilient Underwater Sensor Networks."

[31] L. Vihman, M. Kruusmaa and J. Raik, "Data-Driven Cross-Layer Fault Management Architecture for Sensor Networks," *2020 16th European Dependable Computing Conference (EDCC)*, Munich, Germany, 2020, pp. 33-40, doi: 10.1109/EDCC51268.2020.00015.

[32] Keskonnaagentuur 2023, *Vaatlusandmed*, Keila river water station. accessed: 20.06.2023 www.ilmateenistus.ee/siseveed/vaatlusandmed/tabel/

[33] Keskonnaagentuur 2023, *Vaatlusandmed*, Tallinn-Harku weather station, accessed: 20.06.2023 www.ilmateenistus.ee/ilm/ilmavaatlused/vaatlusandmed/tunniandmed/

Margit Egerer received the B.Sc. degree in engineering physics from Tallinn University of Technology, Tallinn, Estonia, in 2007, M.Sc mechanics of solids and fluids from Chalmers University, Göteborg, Sweden in 2010, M.A. and Ph.D in mechanical and aerospace engineering from Princeton University, NJ, USA in 2010 and 2014 respectively.

She is currently a Researcher with the Centre for Biorobotics, School of Information Technologies, Tallinn University of Technology, where she is involved in the H2020 ILIAD Project. Her research interests include fluid flow sensors, environmental sensing and experimental fluid dynamics.

Asko Ristolainen received the B.Sc. and M.Sc. degrees in mechatronics and the Ph.D. degree in information and communication technology from the Tallinn University of Technology, Tallinn, Estonia, in 2008, 2010, and 2015, respectively.

He is currently a Senior Researcher with the Centre for Biorobotics, School of Information Technologies, Tallinn University of Technology, where he is involved in the H2020 ILIAD Project. His research interests include tactile and flow sensor design in robotics and remote environmental sensing.

Laura Piho received the Master of Mathematics degree in mathematics from the University of Leicester, UK, in 2015, and the Ph.D. degree in engineering from the University of Warwick, UK, in 2019.

She is currently a researcher the Center for Biorobotics, Tallinn University of Technology. During the past years, she was involved in data analysis and processing for flow sensing and bio-inspired robot locomotion. Her research mainly focuses on signal processing, time-series analysis and statistical modeling for the purpose of bio-inspired robotics and environmental sensing.

Lauri Vihman received the B.Sc. and M.Sc. degrees in Computer Science from Tallinn University of Technology in 2013 and 2016.

He is currently an Early Stage Researcher in the Centre for Dependable Computing Systems in Tallinn University of Technology. His research focuses on cross-layer reliability in sensor networks.

Maarja Kruusmaa received her Ph.D.from Chalmers University of Technology in 2002 and since 2008 is a professor in Tallinn University of Technology (TalTech) as a PI of Centre for Biororobitcs, a research group focusing on bio-inspired robotics, underwater robotics and novel underwater sensing technologies. 2017 - 2022 she was a visiting professor in NTNU AMOS (Centre for Excellence of Autonomous Marine Operations and Systems). Her research interests include novel locomotion mechanisms for underwater environments and on flowable media and novel methods for underwater flow sensing.

Curriculum Vitae

1. Personal data

Name	Lauri Vihman
Date and place of birth	21th April 1976 Tallinn, Estonia
Nationality	Estonian

2. Contact information

Phone	+372 505 5575
E-mail	lauri.vihman@taltech.ee

3. Education

2018–…	Tallinn University of Technology, School of Information Technologies, PhD studies
2013–2016	Tallinn University of Technology, School of Information Technologies, Business Information Technology, MSc
2009–2013	Tallinn University of Technology, School of Information Technologies, Business Information Technolgoy, BSc

4. Language competence

Estonian	native
English	fluent
Finnish, German, Greek, Russian	basic

5. Professional employment

2018– …	Tallinn University of Technology, Early Stage Researcher
2016–2018	Zeroturnaround Ltd, Software Engineer
2015–2016	LHV Ltd, Sr. Software Developer
2012–2015	EMT Ltd, System Architect
2009–2012	Zone Media, Software Developer
2002–2009	EMT Ltd, Administrator of Monitoring Systems

6. Defended theses

- 2016, Organizational Sensors: Comparative analysis, Enhancement, Experiment, MSc, supervisor Prof. Innar Liiv, Tallinn University of Technology, Institute of Informatics
- 2013, Generating Memorable Passwords using Character-based Markov Chains, supervisor Dr. Kairit Sirts, Tallinn University of Technology, Institute of Informatics

7. Field of research

- Sensor Networks
- Hardware faults
- Software Architecture

8. Scientific work

Papers

1. L. Vihman, M. Kruusmaa, and J. Raik. Data-Driven Cross-Layer Fault Management Architecture for Sensor Networks. In *2020 16th Eur. Dependable Comput. Conf.*, pages 33–40. IEEE, sep 2020
2. L. Vihman, M. Kruusmaa, and J. Raik. Systematic Review of Fault Tolerant Techniques in Underwater Sensor Networks. *Sensors*, 21(9):3264, may 2021
3. L. Vihman and J. Raik. Adaptive Kalman Filter Based Data Aggregation in Fault-Resilient Underwater Sensor Networks. In *2023 24th Int. Conf. Digit. Signal Process.*, pages 1–5. IEEE, jun 2023
4. L. Vihman, T. M. Parts, H. K. Aljas, M. Thalfeldt, and J. Raik. Algorithms for online CO2 baseline correction in intermittently occupied rooms. In *18th Heal. Build. Eur. Conf.*, Aachen, Germany, 2023. Emerald

1. Elulookirjeldus

1. Isikuandmed

Nimi	Lauri Vihman
Sünniaeg ja -koht	21.04.1976, Tallinn, Eesti
Kodakondsus	Eesti

2. Kontaktandmed

Telefon +372 505 5575	
E-post	lauri.vihman@taltech.ee

3. Haridus

2018–. . .	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, doktoriõpe
2013–2016	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, äriinfotehnoloogia, MSc
2009–2013	Tallinna Tehnikaülikool, Infotehnoloogia teaduskond, äriinfotehnoloogia, BSc

4. Keelteoskus

eesti keel	emakeel
inglise keel	kõrgtase
kreeka, saksa, soome, vene keel	vähene

5. Teenistuskäik

2018– . . .	Tallinna Tehnikaülikool, nooremteadur
2016–2018	AS Zeroturnaround, tarkvarainsener
2015–2016	AS LHV, seenior tarkvaraarendaja
2012–2015	AS EMT, infosüsteemide arhitekt
2009–2012	Zone Media OÜ, programmeerija
2002–2009	AS EMT, monitooringu administraator

Kaitstud lõputööd

- 2016, Organisatsiooni sensorid: võrdlev analüüs, edasiarendus, eksperiment, MSc, juhendaja Prof. Innar Liiv, Tallinna Tehnikaülikool, Informaatikainstituut
- 2013, Meeldejäävate salasõnade genereerimine kasutades täheühenditel põhinevaid Markovi ahelaid , BSc, juhendaja Dr. Kairit Sirts, Tallinna Tehnikaülikool, Informaatika instituut

Teadustöö põhisuunad

- Sensorvõrgud
- Riistvaravead
- Süsteemiarhitektuur

8 Teadustegevus

Teadusartiklite, konverentsiteeside ja konverentsiettekannete loetelu on toodud ingliskeelse elulookirjelduse juures.

ISSN 2585-6901 (PDF)
ISBN 978-9916-80-110-9 (PDF)