Energy-aware distributed computing : application to leak detection in wireless sensor network-based water pipeline monitoring systems

Valery Nkemeni

N°d'ordre NNT : 2021LYSE1150

THESE de DOCTORAT DE L'UNIVERSITE DE LYON

opérée au sein de

l'Université Claude Bernard Lyon 1

en cotutelle avec

l'Université de Buea - CAMEROUN

Ecole Doctorale N° 160

Electronique Electrotechnique Automatique (EEA)

Spécialité de doctorat :

Discipline : Electronique

Soutenue publiquement le 20/07/2021, par :

Valery NKEMENI

Energy-aware distributed computing: Application to leak detection in wireless sensor network-based water pipeline monitoring systems

---

Devant le jury composé de :

LE MOUËL Frédéric	Professeur	INSA Lyon	Président
ANTOINE-SANTONI Thierry	Maître de Conférences HDR	Université de Corse	Rapporteur
EYEBE FOUDA Jean Sire Armand	Maître de Conférences	Université de Yaoundé I, Cameroun	Rapporteur
BELLEUDY Cécile	Maître de Conférences HDR	Université Côte d'Azur	Examinateurice
DUROC Yvan	Professeur	Université Lyon 1	Examinateur
TANYI Emmanuel	Professeur	Université de Buea, Cameroun	Examinateur
MIEYEVILLE Fabien	Professeur	Université Lyon 1	Directeur de thèse
TSAFACK Pierre	Chargé de Cours	Université de Buea	Co-directeur de thèse

Dedication

To my daughter Kenisha

Acknowledgements

I wish to express my profound gratitude to my supervisors, Prof. Mieyeville Fabien and Dr. Tsafack Pierre. I am grateful for their guidance throughout my Ph.D. journey. It was an exciting experience working with such talented mentors, and I learned a lot from them.

I wish to sincerely thank the Cooperation and Cultural Action Service (SCAC) of the French Embassy in Cameroon for their financial support during my PhD studies. I am also thankful for the research facilities and support of the Ampere laboratory of l'Université Lyon 1 and the Faculty of Engineering and Technology of the University of Buea, that permitted me to successfully conduct my experiments, present my results at an international conference, and publish my work in an international peer-reviewed journal.

I am grateful to Prof. Tanyi Emmanuel, the dean of the Faculty of Engineering and Technology (FET), for all his tireless efforts in starting the postgraduate programme at FET that led to my Ph.D. admission.

I wish to thank all the professors, researchers, Ph.D. students, lab technicians, and administrative assistants of both Ampere laboratory and FET for their advice and assistance during my Ph.D. studies.

I am thankful to Dr. Fendji Marie Danielle, Dr. Ngassam Ines, and Dr. Djoupo Gaston for proofreading the draft of the extended summary of my thesis in French.

Finally, I am infinitely grateful to my mother, Mrs Seme Martina and my wife, Mrs Nyenti Sophie for all their encouragement and support during my PhD studies.

Résumé

L'eau est un besoin fondamental pour la vie humaine, notamment pour la boisson et les autres activités domestiques. Malheureusement, de grandes quantités d'eau potable traitée se perdent chaque jour de par le monde et plus particulièrement dans les pays en développement à cause de fuites dans les réseaux de distribution d'eau. Depuis plus d'une décennie, des systèmes de surveillance des canalisations d'eau basés sur des réseaux de capteurs sans fil – Wireless Sensor Network-based Water Pipeline Monitoring (WWPM) – sont utilisés pour détecter les fuites et permettre une intervention rapide pour les réparer. Cependant, la plupart des solutions de surveillance des canalisations d'eau sont confrontées à des problèmes tels que la difficulté de détecter les fuites en temps réel, le coût élevé et la difficulté d'installation. Récemment, les accéléromètres à faible coût conçus à base des systèmes micro-électromécaniques (MEMS) ont gagné en popularité dans les systèmes WWPM en raison de leur faible coût, de leur faible consommation et de leur caractère non intrusif qui les rend faciles à installer. Cependant, la précision des systèmes WWPM utilisant des accéléromètres MEMS est faible et doit être améliorée.

L'objectif général de cette étude est d'augmenter la précision des mesures et la durée de vie des capteurs du reseau sans fil grâce à une approche distribuée. L'objectif spécifique est de proposer une solution WWPM en temps réel et à faible puissance utilisantdes accéléromètres MEMS à faible coût et un filtre de Kalman distribué (DKF) pour améliorer la précision de la détection des fuites. Pour atteindre une surveillance en temps réel, nous avons proposé une solution entièrement distribuée en mettant en œuvre un traitement distribuée de données dans le système WWPM; ici tout le traitement nécessaire à la détection des fuites est effectué en utilisant les ressources informatiques embarquées dans les nœuds de capteurs, sans nécessiter de communications multi-sauts vers une station de base. Pour améliorer les performances de détection des fuites tout en préservant la durée de vie du système WWPM, nous avons propose distribution des données fusionnéesen mettant en œuvre un DKF dans le système WWPM. Enfin, pour assurer à la fois une surveillance en temps réel et une faible consommation d'énergie, nous avons mis en œuvre une détection hiérarchique et un cycle de travail au niveau des capteurs. Dans cette thèse, nous avons commencé par pour un capteur capable de traiter in-situ sous une faible contrainte énergétique. Cela nous a conduit à la sélection du microcontrôleur ESP32, de l'émetteur-récepteur nRF24L01+ et des accéléromètres LSM9DS1

et ADXL344 comme composants de notre capteur. Nous avons ensuite développé une solution WWPM qui utilise le calcul distribué en mettant en œuvre un DKF au sein du système WWPM. Les résultats des simulations et des expériences en laboratoire ont révélé que la propriété de distribution des données fusionnées du DKF a augmenté la précision de la détection des fuites et a également préservé la durée de vie du WWPM. En outre, nous avons évalué les performances de détection des fuites et la consommation d'énergie de trois DKF, choisis parmi les stratégies de fusion de données distribuées basées sur la diffusion, le bavardage et le consensus. Les résultats de simulation et de laboratoire ont révélé que le DKF basé sur le consensus avait la meilleure précision de détection des fuites, tandis que le DKF basé sur la diffusion avait la plus faible consommation d'énergie. Enfin, nous avons terminé l'étude en mettant en œuvre la détection hiérarchique et le cycle de travail sur chaque nœud en plus du calcul distribué. Les résultats ont montré une diminution significative de la consommation d'énergie tout en permettant une détection des fuites en temps réel.

En résumé, dans cette thèse, nous avons démontré que l'approche distribuée fonctionne bien et présente de réels avantages. La plupart des études de la littérature démontrant la faisabilité de l'approche distribuée se limitant à des simulations, nous avons validé le concept de manière expérimentale et proposé une approche complète qui inclut la conception, la simulation et le déploiement physique d'un réseau de capteurs, en tenant compte des enjeux énergétiques.

Mots clés : calcul distribué ; réseaux de capteurs sans fil ; filtre de Kalman distribué ; surveillance des canalisations d'eau ; détection des fuites ; accéléromètres MEMS à faible coût, consommation d'énergie ; fusion de données distribuée.

Abstract

Water is a basic necessity for human living, especially for drinking and other domestic and industrial activities. Sadly, in developing countries, large quantities of treated drinking water get lost daily through leaks on the water distribution networks. For over a decade now, wireless sensor network-based water pipeline monitoring (WWPM) systems have been used to detect leaks and prompt rapid repair intervention. However, most WWPM solutions face challenges, such as difficulty in achieving real-time leak detection, high cost, and installation difficulty. Recently, low-cost micro-electro-mechanical systems (MEMS) accelerometers have gained popularity in WWPM due to their low-cost, low-power and non-intrusive feature which makes them easy to install. However, the accuracy of WWPM systems employing low-cost MEMS accelerometers is low and needs improvement. The purpose of this study is to investigate a real-time, and low-power WWPM solution that employs low-cost MEMS accelerometers and a distributed Kalman filter (DKF) to improve leak detection accuracy. The goal is to achieve an increase in measurement accuracy and network lifetime via a distributed approach. We began by providing specifications for a sensor node which is capable of in-situ processing under low energy constrain. This led us to the selection of the ESP32 microcontroller, nRF24L01+ transceiver, and LSM9DS1 and ADXL344 accelerometers as the components of our custom sensor node that was used for pipeline monitoring. We then developed a WWPM solution that employed distributed computing by implementing a DKF within the WWPM system. The results from both simulations and laboratory experiments revealed that the distributed data fusion property of the DKF increased the leak detection accuracy and also preserved the WWPM lifespan. Furthermore, we evaluated the leak detection performance and energy consumption of three DKF algorithms, selected from distributed data fusion strategies based on diffusion, gossip, and consensus. The simulation and experimental results revealed that the consensus-based DKF had the highest leak detection sensitivity while the diffusion-based DKF had the lowest energy consumption. Finally, we ended the study by implementing hierarchical sensing and duty cycling on each node alongside distributed computing. The results showed a significant decrease in the energy consumption while achieving real-time leak detection. Summarily, in this thesis we demonstrated that the distributed computing works well in wireless sensor networks (WSN) and has real advantages. With most studies in the literature demonstrating the feasibility of distributed computing in WSN being limited to simulations, we

validated the concept experimentally and proposed a complete approach that includes the design, simulation, and physical deployment of a WSN, taking into account the energy issues.

Keywords: distributed computing; wireless sensor networks; distributed Kalman filter; water pipeline monitoring; leak detection; low-cost MEMS accelerometers, power consumption; distributed data fusion

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the applications and constraints of Wireless Sensor Networks (WSNs) and the need for shifting towards a distributed architecture in WSN applications. We briefly discuss Water Pipeline Monitoring (WPM) with a focus on the challenges of worldwide water distribution and in particular water distribution in developing countries; we also briefly discuss the need for reliable leak detection systems for WPM. The motivation for a distributed computing solution for WPM using WSN is also presented. The research objectives and the original contribution to knowledge are discussed, followed by the scope of the study and the organization of the thesis.

1.1 Towards a Distributed paradigm in Wireless Sensor Networks

1.1.1 Applications of WSN

A Wireless Sensor Network (WSN) consists of several embedded nodes with sensing, processing, and wireless communications capabilities, distributed over an area of interest to monitor physical or environmental conditions [1]. Being spatially distributed systems, WSNs exploit wireless communication as the means of communication between nodes. This makes them effective for a myriad of applications.

Application areas of WSNs include geographical monitoring, habitat monitoring, transportation, military systems, business processes, structural health monitoring, microclimate research, medical care and others [2]. Kandris et al. [3], in an up-to-date survey on the applications of WSN, classified the applications of WSN into six main categories (military, environmental, health, flora and fauna, industrial, and urban) based on the nature of their use as shown in Figure 1.1. The figure also depicts the sub-categories of each main category of WSN applications described in [3].

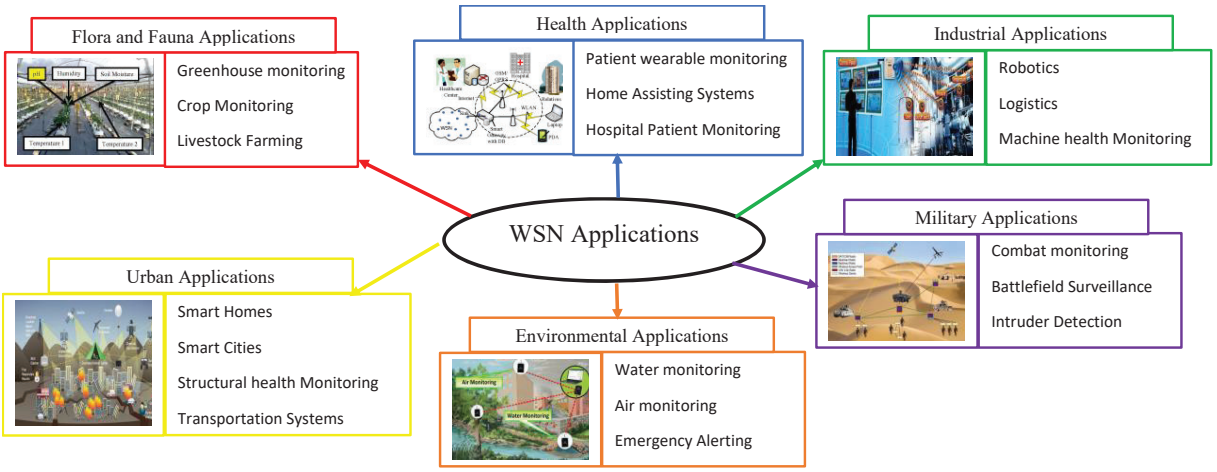

Figure 1. 1: Application areas for WSN. Adapted from [3]

1.1.2 Constraints of WSN

A wireless sensor node is typically a compact sized hardware unit that acquires desired data from the environment and communicates wirelessly to other nodes in the network to relay the raw data gathered or extracted information to a central data sink. Its major components include: a communication unit, a control and processing unit, a memory unit, a power supply unit, and a sensing unit [4]. Thus, a WSN may consists of nodes which have the potential to sense, compute and communicate. However, wireless sensor nodes are inherently resource constrained, usually having limited processing capability, storage capacity, and communication bandwidth. These limitations are partly due to the two greatest constraints i.e., limited energy and physical size [5].

Over the years, due to technological advances, the hardware architecture of wireless sensor nodes has evolved from first-generation sensor nodes (e.g. Tmote Sky, MicaZ, Mica2, Micadot, etc.) that use of 8-bit microcontrollers, through second-generation sensor nodes (e.g. TelosB) that incorporate 16-bit microcontrollers such as the MSP430, and finally to third-generation sensor nodes that employ 32-bit microcontrollers (e.g. 32-bit microcontrollers based on ARM Cortex – M0/M0+/M3/M4/M7, dual-core ESP32, and PIC32MZ) [6], [7]. The first- and second-generation sensor nodes performed little or no local processing due to the constrain in their onboard computing power and memory. However, the third-generation sensor nodes have a significant onboard computing power and memory that can achieve in-situ processing.

1.1.3 Shifting towards a distributed computing approach in WSN

Most WSN monitoring applications in the literature and especially in WPM are centralized [8]–[10]. This is because historically, the early sensor nodes acted as mere data collectors and wireless relays due to their low onboard computing power that could only enable them to sense and communicate data that was more or less pre-processed. This leads to the underutilization of the processing unit and overutilization of the communication unit of sensor nodes since their primarily role in such centralized architectures is to collect and transmit data periodically to an intelligent central base station, where all the processing is done to detect anomalous behaviours [8], [11], [12].

In a large-scale WSN, one of the main objectives is to achieve low power consumption, to enable the sensor nodes to be operational for long periods without replacing their battery since the sensor nodes in such applications are usually battery-powered and often inaccessible physically [13]. The drawbacks of WSNs with centralized architectures deployed in large-scale monitoring applications include enormous bandwidth requirements and high energy consumption. This is because periodic transmission of raw data over long distances, via multiple hops, to the base station leads to fast depletion of sensor node's battery and shortens the lifespan of a WSN [12], [14], [15], since research has shown that the communication unit consumes the greatest portion of a sensor node's energy [1], [16], [17]. Thus, it is obvious that communication is a considerably more energy-consuming venture than computation. To reduce the energy consumption of sensor nodes in a WSN, a wise thing to do is to invest more in computation within the WSN, whenever possible to save on communication costs. Therefore, by minimizing the amount and range of communications as much as possible through local collaboration among sensor nodes, one can significantly prolong the lifespan of a WSN. Finally, other drawbacks of centralized WSN applications include low reliability and robustness, longer response time, and low-level data security and privacy [12], [14], [15], [18], [19]. These disadvantages of centralized WSNs have led to active research in recent years directed towards distributed computing in WSNs.

Recently, some works in the literature have demonstrated via simulations the feasibility of distributed computing in WSN and its promises of achieving a gain in performance and lowering of the power consumption [9], [12], [18]–[20]. By performing more local computation, limiting

exchanges only between neighbouring nodes, and reducing the number of messages that need to be transmitted, distributed computing in WSN has the potential of providing a solution to the drawbacks of the centralized approach [18], [20]–[22].

1.2 Water Pipeline Monitoring

1.2.1 Structure of Water Supply Systems

Water is a basic necessity for everyday life, required for many human activities such as drinking, irrigation of crops, recreational activities, and for the effective accomplishment of many industrial processes [23]. In most communities, water transportation via pipelines to users seems to be the most economical way [10] and consists of water supply systems comprising of two main parts (Figure 1.2): (1) The transmission mains, which are pipes responsible for transporting water to tanks and (2) Water Distribution Networks (WDN) , which are pipes and service connections for distributing water to users. These infrastructures are usually not completely watertight as even in the most recent and well-built WDN, some level of leakage and occasional pipe bursts occur, leading to water losses [24].

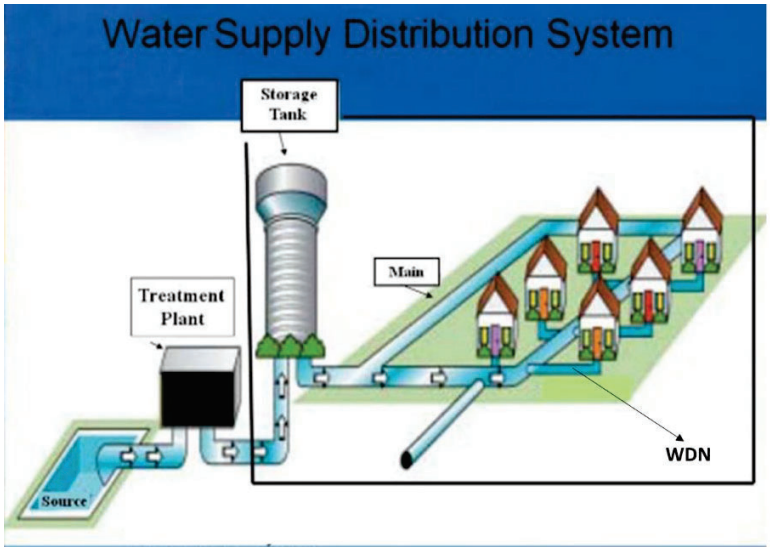

Figure 1. 2: Water supply system

1.2.2 The Problems of Water Distribution in Developing Countries

Water pipeline leakages are one of a few problems faced by water utility companies all over the world as water loss through leakages is recognized as a costly problem worldwide, due to the waste of precious liquid, as well as from the economic point of view [25]–[27]. According to a publication by UNICEF in 2015 [28], the data revealed that more than 30% of the developing and less developed countries have no access to quality drinking water sources. Additionally, another report published by the World Bank in 2016 [29], indicated that in developing countries, roughly 45 million cubic meters of water are lost daily with an economic value of over US $3 billion per year. The report also stated that saving half of those losses would provide enough water to serve at least 90 million people. In Cameroon, a sub-Saharan developing country, the level of Non-Revenue Water (NRW), which is the portion of the total amount of water produced for which the water utility company generates no income (because it is lost via leaks/burst and/or theft) is at 4.67% [30], [31]. Recently, the paper [32] revealed that the global volume of NRW is 346 million cubic meters per day. The high level of NRW is due to aging WDN infrastructures that create physical losses through leaks or bursts.

1.2.3 The Need for Reliable Real-time Leak Detection Systems

Water demand is increasing continuously and rapidly as a result of the growth of the Earth's population, but water resources are facing a constant decrease caused by global heating and climate change [25], [33]. Unlike other more peculiar phenomena, water scarcity is common to both developing and developed countries [25]. The scarcity of water thus requires that water losses resulting from leaks be minimized by accurately detecting and localizing leakages in real-time, each time they occur. Furthermore, since the occurrence of leakages expose treated water to the external environment, it is necessary to protect treated drinking water from contamination (which may lead to it serving as a potential avenue for the outbreak of diseases) [23], by quickly identifying leaks in the WDN when they occur. Over the years, this has led to numerous research studies in the field [34]–[47], providing a wide range of methods for detecting and locating leaks on water pipelines.

1.2.4 WSN-based Water Pipeline Monitoring Systems

WSN-based Water Pipeline Monitoring (WWPM) systems consist of a number of sensor nodes that periodically collect leak signals from the pipe. The signals are then processed to detect the presence or absence of a leak on the pipeline. Given that WDNs are linear structures, deploying a WWPM solution as a linear WSN is much feasible. However, most WWPM solutions in the literature have neglected the linear WSN design [10]. Even though very few WWPM studies have deployed a linear topology [10], we will focus in this study on the deployment of WWPM systems as a linear WSN.

There are several factors affecting pipeline monitoring schemes such as communication mechanism, evaluation methods, power management, monitoring types, sensor connectivity, sensing coverage, sensing methods, and types of sensors [10]. Notwithstanding, the biggest challenge with leak detection in WWPM systems using low-cost sensors is that the leak signals may be inaccurate due to the low sensitivity of the sensors and environmental noise, and may result in false alarms in the leak detection system. Thus, the issue of reliably identifying a leak signal in the midst of errors from a number of sources (commonly called noise) is a fundamental challenge of any leak detection system [46], [48].

Depending on where the leak signals from remote sensors are processed, WWPM solutions can be classified as either centralized, decentralized or distributed. Several studies have proposed centralized WWPM solutions in the literature [41], [42], [46], [49], [50]. In these studies, the sensor nodes periodically measure leak signals from the pipe where they are installed and transmit to a central base station (where the leak detection algorithm is found) for further processing in order to detect the presence of a leak on the pipeline. Such centralized schemes are characterized with a large number of multi-hop transmissions and may deplete the sensor node's energy faster. Furthermore, some studies [13], [38], [40], [51] have proposed decentralized solutions where some processing for leak detection is performed at the sensor node. However, to the best of our knowledge, no study has proposed a fully distributed solution where all the processing required for leak detection is performed by the sensor node.

1.3 Statement of the Problem

1.3.1 Context

For more than a decade now, WWPM systems have been popularly used for monitoring of WDNs. However, existing WWPM solutions face a lot of the challenges such as high energy consumption and difficulty in achieving real-time leak detection, since they are centralized and also because they employ intrusive (pressure and flow) sensors that are costly, difficult to install, and consume more energy. In recent years, WWPM systems using vibration sensors have become popular [35], [43], [46]. Vibration sensors can be used for monitoring because water pipeline pressure monitoring can be transformed to vibration monitoring of the pipe surface [37], [52], [53], since a transient change in pressure when a leak occurs is always accompanied by an increase in the pipe surface acceleration at the corresponding locations along the pipe length [53]. The pressure fluctuations in a pipe are thus related to the pipe surface vibration via a nonlinear but proportional relationship [37], [53]. Vibration sensors (accelerometers, piezoelectric transducer, force sensitive resistors, etc.) are easy to install, less costly to maintain/operate, and consume less energy. In addition, the sensors offer the distinct advantage of providing real-time monitoring of the WDN which can prompt immediate interventions [46].

1.3.2 Problem

In most developing countries, the WDN is made up of plastic pipes and it has been shown in the literature that the propagation of leak signals (vibration) does not go far in plastic pipes [54], [55]. Thus, reliable leak detection will require the sensors to be placed very close to each other to have a higher spatial resolution [56]. High accuracy accelerometers attached to the outer surface of the pipe can be used to accurately detect this sudden increase in the pipe surface acceleration caused by leaks on the pipeline. However, the need for lower inter-sensor distances and the expensive nature of such accelerometers will increase the overall cost, which makes them not suitable for installation in developing countries. Therefore, low-cost MEMS accelerometers can be a feasible and economically viable solution for deployment in developing countries. While previous WWPM studies using low-cost MEMS accelerometers have been useful in detecting leakages, they still have the challenge of reliably detecting leaks amid random environmental noise due to the low accuracy of the sensors [35], [46], [57], [58]. Another challenge is the inability of these previous

WWPM systems to provide reliable real-time leakage detection while preserving the WSN lifetime since most of them are centralized. However, since WWPM systems employ multiple low-cost sensors to monitor the pipeline, multi-sensor data fusion techniques, which have been successfully used in target tracking applications [59], can be used to increase the reliability of leak detection systems that use low-cost vibration sensors [57].

1.3.3 Proposed Solution and Benefits

Multi-sensor data fusion can combine redundant data from multiple low-cost sensors to achieve a more accurate information whose quality exceeds that achieved by using a single sensor [57], [60]. Moreover, the low energy consumption requirement and the need for a WWPM system to go unattended for a long period of time without any replacement of the sensor node's battery [46], [61], [62], affects the choice of a multi-sensor data fusion technique that can be used. Thus, multi-sensor data fusion in WSN can either be done via a centralized, decentralized or a distributed manner [59], [63]. The centralized data fusion technique will require multi-hop communications, which have a higher likelihood of developing an energy hole in the network, thus shortening the lifespan of the WSN [18], [21], [22]. However, using distributed data fusion may increase the lifespan of the WSN as there is no central point for fusion, and multi-hop communications will be eliminated entirely. The objective of distributed data fusion is to use distributed computations across the network such that the local information at each sensor node converges to an optimal value [64].

Implementing distributed data fusion in WWPM systems composed of a network of low-cost MEMS accelerometers will provide the following benefits:

- It will enable the realization of real-time monitoring, permitting leaks to be detected as they occur. This will lead to fast intervention, thereby reducing the quantity of treated water that will be lost and prevention of the contamination of treated drinking water.
- It has the capability of reducing the power consumption of the sensor nodes while preserving the WSN lifetime, thereby enabling the WWPM system to be operational for a long period without needing battery replacement. This is because processing is done at the node level and the number of multi-hop transmission is reduced. This reduces the energy spent on communication.

- It will reduce the bandwidth requirement of the WSN since processing is done by the sensor node's processing unit, and sensor nodes will only communicate with neighbouring nodes to achieve some higher accuracy. This is advantageous since WSNs are bandwidth constrained.

Inspired by the application of low-cost MEMS accelerometers in WWPM systems and the challenges of WWPM (such as real-time monitoring, low-cost and low-power consumption requirements) that have not been dealt with properly in previous studies, motivated us to conduct this study. In this study, we demonstrate the viability of Distributed Kalman Filter (DKF), which is a Kalman filter with distributed data fusion, as a distributed computing solution for leak detection in WWPM systems using low-cost MEMS accelerometers. Thus, the proposition of a fully distributed solution based on DKF that is real-time, low-power, and reliably detects leaks in WWPM systems using low-cost MEMS accelerometers will be the focus of this thesis.

1.4 Research Aim and Objectives

The main aim of this thesis is to establish an increase in the performance and the lifespan of a WSN via the use of a distributed computing approach. With WWPM being an application domain of WSN that requires low energy consumption [61], the challenge is how to achieve both high leak detection accuracy and maximum service lifetime of the WWPM system. Additionally, WWPM systems present a good application scenario where efficient distributed computing can be demonstrated in WSNs (due to their 1D linear topology which makes them easier to manage for a first demonstration). Thus, the purpose of this study is to investigate the application of DKF in achieving low-power, real-time, and reliable leak detection in WWPM systems using low-cost MEMS accelerometers.

The aim of this research will be achieved by meeting the following specific objectives:

1. Review popular water pipeline monitoring techniques available in the literature with a focus on WWPM solutions that use low-cost vibration sensors for monitoring. From there, identify the knowledge gaps in the field of leak detection in plastic WDNs using WWPM systems composed of low-cost MEMS accelerometers.

2. Establish specifications for a low-cost and low-power sensor node capable of in-situ processing under energy constraints. The sensor node should have sufficient computing capacity to perform local processing and be energy-aware. On this basis, design and implement a wireless sensor node based on the established specifications.
3. Propose a fully distributed, real-time, and low-power WWPM solution based on DKF for reliable leak detection in plastic pipes. In this regard, perform a thorough literature search on DKF algorithms that can be implemented in WSNs and select three DKF algorithms whose leak detection performance and power consumption will be evaluated in the context of leak detection in WWPM systems using low-cost MEMS accelerometers.
4. Demonstrate the feasibility of applying a DKF for reliable real-time leak detection in a WWPM system that uses low-cost MEMS accelerometers by deploying a sensor network that implements one of the selected DKF algorithms. On this basis, evaluate the leak detection performance and energy consumption of the proposed distributed solution by performing simulations and validating the simulation results using results obtained from physical experiments on a laboratory testbed.
5. Evaluate the leak detection performance and power consumption of the three selected DKF algorithms using both simulations and physical experiments. On this basis, compare the leak detection performance and power consumption of the selected DKF algorithms and propose which of the DKF is optimal for implementation in WWPM systems using low-cost MEMS accelerometers.

1.5 Contributions

While extensive research has already been conducted on WWPM systems using low-cost vibration sensors, several challenges still exist, such as providing a leak detection system that is fully distributed, real-time, low-cost, and consumes less power. Besides, the implementation and evaluation of fully distributed leak detection algorithms applicable for real-world WSNs and that use of distributed approaches are rare [10]. Based on the knowledge gaps in the literature, this work is novel and it is the first application of DKF in the context of leak detection in WWPM systems using low-cost MEMS accelerometers.

This thesis has made some research contributions that not only improve on the state-of-the-art of real-time leak detection in WWPM systems using low-cost MEMS accelerometers but that of linear WSN applications as well. They include:

1. The proposition, design, and deployment of a low-cost custom sensor node based on low-cost commercial off-the-shelf components (COTS) such as ESP32 microcontroller (processing unit), nRF24L01+ transceiver (communication unit), and ADXL344 and LSM9DS1 accelerometers (sensing unit).
2. The demonstration of a reliable, real-time, fully distributed, and DKF-based solution for leak detection in plastic WDNs using WWPM systems composed of low-cost MEMS accelerometers.
3. The evaluation of the performance and power consumption of the proposed DKF-based leak detection solution via simulations in CupCarbon 4.2 simulator and validation on an experimental testbed. The comparison of simulation results and experimental results is a key contribution for the real-life implementation of the proposed approach; since there are many studies in the literature that do not get to experimental analysis and the study of the behaviour of the sensor nodes under real-life conditions. The results showed that the distributed data fusion capability of the DKF improves the reliability of leak detection while preserving the WSN lifetime.
4. The evaluation of the leak detection performance and power consumption of three selected DKF algorithms via simulation and validation on an experimental testbed. The results showed that diffusion-based DKF algorithms consume less energy (preserve the WSN lifetime) but have lower leak detection reliability whereas, consensus-based DKF algorithms consume more power but have a higher reliability.
5. The implementation of duty cycling and hierarchical sensing at the sensor node level to reduce the power consumption of the proposed fully distributed leak detection solution.

Summarily, the original contribution to knowledge of this thesis is the demonstration and evaluation of a low-cost, low-power, and fully distributed solution based on DKF for real-time leak detection in WWPM systems using low-cost MEMS accelerometers.

1.6 Thesis Scope

Given that the Kalman filter can be applied as a state observer and data fusion algorithm, we focus in using the Kalman filter in this study as a sensor fusion algorithm to improve on the accuracy of leak measurements collected by local sensor nodes. As a state observer technique, the Kalman filter is used to compute an optimal estimate of an interested parameter (e.g., leak size and location) given only limited available information (e.g., pressure and flow) whereas as a sensor fusion algorithm, the Kalman filter is used for optimal state estimation of a parameter based on data from dynamic models and measurements from sensors or the fusion of measurements from multiple sensors. Kalman filter methods used for leak detection in WDN have largely been applied to pressure and flow measurements. In the literature, the Kalman filter and its variants have been used as model-based techniques for leak detection with flow and pressure measurements from intrusive sensors serving as data sources. However, we use the Kalman filter in this thesis as a signal processing algorithm acting on measurements from MEMS accelerometers and performing data fusion so as to isolate leak signals from noise. Additionally, we investigate only DKF algorithms that can be implemented on wireless sensor nodes.

1.7 Thesis layout

This thesis contains 7 chapters and is organized as follows:

- Chapter 2 presents a background on energy management techniques and distributed computing in WSNs. It also provides a comprehensive literature review on WWPM solutions and the gaps in knowledge.
- Chapter 3 presents the selection of COTS components used for the design and implementation of a custom sensor node that will be deployed on the laboratory testbed.
- Chapter 4 presents a survey of DKF algorithms for implementation in WSN and the selection of three DKFs that will be implemented and whose leak detection performance and power consumption will be evaluated in the context of leak detection in plastic WDNs using WWPM systems composed low-cost MEMS accelerometers.

- Chapter 5 presents the results of an initial demonstration of the application of DKF in the realization of a fully distributed, real-time, reliable, and low-power leak detection solution for WWPM systems using low-cost MEMS accelerometers.
- Chapter 6 presents the results of the evaluation of the leak detection performance and power consumption of the three selected DKF algorithms for experiments conducted on a simulation platform and a laboratory testbed. It also presents the results obtained from experimentally implementing of duty cycling and hierarchical sensing on the sensor nodes.
- Chapter 7 presents a number of conclusions derived from the achievement of the objectives of this study and also provides recommendations for future works.

Chapter 2

Distributed Computing and Wireless Sensor-Based Water Pipeline Monitoring

This chapter presents the background and literature review of our study. It starts by classifying WSN monitoring applications into centralized, decentralized, and distributed, based on where data collected within the WSN are processed. This is then followed by a discussion on the drawbacks of centralized WSN monitoring applications. Next, the power management techniques available in the literature for maximizing WSN lifetime are later reviewed. The power management techniques are organized under energy harvesting, energy balancing, and energy conservation techniques. From there, a discussion on distributed computing in WSN and its benefits is then presented. A number of implementations of distributed computing in WSN, categorized under distributed query processing, collaborative signal processing, distributed state estimation/detection, and in-network processing are later discussed. We later move on to present water pipeline monitoring with a focus on WWPM systems using low-cost vibration sensors. Mention is made of the challenges of vibration-based leak detection methods in plastic pipes and the need for improving the reliability of leak detection. We also review WWPM solutions available in the literature and classify them as either centralized, decentralized, and distributed depending on where the pre-processing, leak detection and leak localization algorithms are executed. This is followed by a review of WWPM studies that applied energy conservation techniques to extend the WWPM system's lifespan. Lastly, we present a summary of the identified gaps in knowledge.

2.1 Classification of WSN Monitoring Applications

There exist several criteria for classifying WSNs and hence several taxonomies. However, we are interested in the criterion that puts data and its management at the heart of the problem. Hence, the criterion chosen for our classification is based on data processing. The classification of WSNs based on data processing determines the location where data processing algorithms are implemented (sensor node or base station). It also determines whether the base station receives

raw data, processed data or decisions from the sensor nodes [65]. Additionally, it affects the computation and communication costs and also influences the scalability of the network and the kind of WSN applications where it can be applied [63]. In general, depending on the technique of data processing, WSN monitoring applications can be classified into three categories [40], [59], [60], [63], [65]:

1. Centralized monitoring
2. Decentralized monitoring, or
3. Distributed monitoring

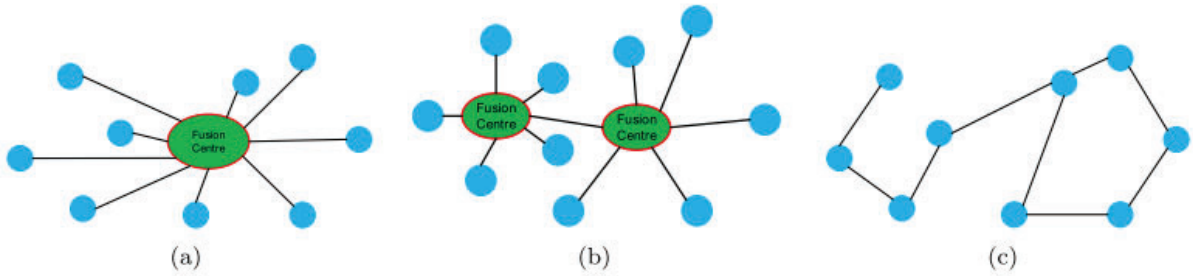

Figure 2. 1: WSN monitoring schemes: Centralized, Decentralized, and Distributed **[59]**

Figure 2.1 depicts the different WSN monitoring architectures based on the implemented data processing strategy. The circles denote the sensor nodes while the solid lines denote the communication links between sensor nodes.

In centralized monitoring applications, there is a single global fusion centre [63] called the base station or sink that receives and processes the raw data collected and transmitted by all the sensor nodes in the WSN [59]. This implies that, most, if not all the intelligence is found at the base station and the sensor nodes have the sole responsibility of sensing certain phenomena of the physical world and transmitting them to this central site for processing. In such cases, there is little or no intelligence at the sensor node level and the sensor nodes act primarily in a sense-only fashion [66]. The presence of this single fusion centre increases the number of data transfer and reduces the scalability and energy efficiency of centralized WSNs. This is due to the fact that when a large number of nodes communicate with the base station, the network energy consumption is greater than when only a few nodes transmit their local decisions to the base station [40]. Nevertheless,

the centralized approach has the advantage of being Bayesian optimal in terms of performance, since all measurements are processed at single site.

To utilize the processing power of sensor nodes for data processing at the node level, there are two ways to go about it, which include: the decentralized approach and the distributed approach. In the decentralized approach, local fusion centres (cluster heads) receive raw or pre-processed data from sensor nodes in a cluster and carry out further processing of the raw or pre-processed data, whereas in the distributed approach there is no need for local fusion centres as each node performs processing of its local data and only communicates with its neighbours to reach some desired accuracy [63].

The decentralized approach is a compromise between completely centralized and distributed versions of WSN monitoring scheme. It is also referred to as hierarchical data processing strategy [65] and consists of several fusion centres that are capable of communicating with their close neighbours or directly with the base station (global fusion centre). The sensor nodes in a decentralized WSN are grouped into clusters and each cluster head serves as a local fusion centre that the member nodes within the cluster communicate with. The local fusion centres process the raw data (local measurements) received from sensor nodes within the cluster and transmit it to the base station. Since the local fusion centres perform some processing on the raw data received from the sensor nodes and given that only fusion centres can communicate with the base station, the communication overhead is reduced, thereby improving the energy efficiency. However, the decentralized monitoring schemes are also limited by scalability in the case of large scale WSN [59].

The distributed approach, also referred to as parallel data processing strategy [65], entails that each sensor node performs the data processing on its local measurements [59] without needing fusion centres [63]. In addition, the sensor nodes may communicate with their close neighbours to reach some desired accuracy. The absence of fusion centres in this approach increases its scalability and reduces communication costs [63]. This makes it suitable for applications requiring features such as real-time data capture, processing, and dissemination [40]. Furthermore, it has advantages such as increased robustness and reliability due to its built-in redundancy [40], [59]. As a disadvantage,

if not well designed, it may require a large number of iterative communications in order to achieve an accuracy comparable the centralized approach.

Finally, it should be recalled that the base station in a centralized WSN receives raw data from all the sensor nodes, that of a decentralized WSN receives processed data from the local fusion centres, while that of a distributed WSN receives decisions from the sensor nodes.

Now that we are done classifying WSN monitoring applications based on the underlying data processing strategy, we will move on to discuss the disadvantages and drawbacks of the centralized approach (which is predominantly used in most WSN monitoring applications).

2.2 Drawbacks of Centralized Data Processing in WSNs

When designing WSNs, the basic requirements for an efficient WSN are scalability, fault-tolerance, and energy efficiency. In this section, we will discuss the drawbacks of centralized WSN monitoring applications in the context of these three requirements for efficient WSN design. The drawbacks of centralized WSNs include [15], [18], [63]:

- Lack of scalability
- Low energy efficiency
- Increased latency
- Reduced robustness and reliability
- Low data privacy and security

2.2.1 Lack of scalability

According to Abdulkarem et al. [65], network scalability is the ability of a WSN to permit the addition of new sensor nodes, displacement of sensor nodes within the WSN, and the exit of existing sensor nodes while maintaining the performance and operation of the network. In the case of large-scale WSN, it is required that the WSN covers a large geographical area for monitoring. As such, the network architecture and data processing strategy play an important role on the scalability of the WSN by ensuring that it can be enlarged [65]. Given that in centralized WSN applications, data processing is performed only at the base station, the scalability of centralized WSNs is greatly reduced as the base station cannot effectively communicate with all sensor nodes in a large-scale WSN because of physical constraints such as limited communication range of

sensor nodes, increased communication delay resulting from multi-hop communications, low reliability due to loss of packets, and limited communication bandwidth [59]. This makes the centralized approach not suitable for implementation in large-scale real-time monitoring systems.

2.2.2 Low energy efficiency and reduction in WSN lifetime

Most sensor nodes in large-scale monitoring applications are far from the base station and will require much energy to transmit their data via a single hop to the base station. In most cases, the transmission of the sensor node's data to the base station via multi-hop communications is employed. These multi-hop communications directly increase the number of data transmissions, which also increases the overall power consumption and thus shortens the lifespan of a WSN by decreasing the lifetime of every node serving as a relay. Furthermore, in the centralized approach, nodes directly connected to the base station are involved in relaying all the messages directed to the base station. This results in what is referred to as the energy hole effect [18], [21], [22], which is caused by the uneven distribution of the energy consumption among the nodes in the network. Thus, the energy of nodes directly connected to the base station depletes faster and when they finally get exhausted, the remaining one-hop neighbours of the base station face an even greater burden which quickly disables the network [18]. The existence of an energy hole in the WSN makes it difficult for data from sensor nodes to be delivered to the sink, resulting in a premature end of the WSN lifetime. Therefore, given that energy consumption in a WSN is proportional to the frequency of data transmission, the large number of data transmission involved in the centralized approach makes it not effective for prolonging WSN lifetime.

2.2.3 Increased latency

In large-scale WSN, the sensor nodes at the periphery of the network will have to involve in multi-hop communications in order to get their sensed data to the base station. This multi-hop communications will induce delays thereby increasing the latency. Besides, there is a higher likelihood that some of the packets will be lost, which will further increase the latency via retransmission, since the communication links in WSNs are not very reliable. Furthermore, the centralized approach is not suitable for time-critical applications where the response time is required to be low. This is because decisions can be delayed due to the round-trip transmissions involved [15], [18], [67]. Finally, the limited bandwidth available in WSNs and the larger number

of data transmissions involved in the centralized approach will further increase the delays in the WSN. This makes the centralized approach not suitable for applications requiring real-time data capture, processing, and detection of events.

2.2.4 Reduced robustness and reliability

Fault tolerance of a network is a measure of its ability to do the intended job if some node(s), link(s) or both fail [8]. Centralized WSNs have a single point of failure and are less robust to node failures. This is because a complete reorganisation of the network is required each time a node fails or a node is added to the existing network [68]. In addition, there is a higher likelihood of packets being lost in the centralized approach due to the large number of data transmissions involved. This increase in packet lost is partly due to the fact that the sensor nodes are not very reliable since they are developed from cheap off-the-shelf components. Besides, the energy constrain nature of the nodes necessitates that they regularly go to sleep in order to reduce power consumption, which may also result in loss of packets when the nodes are at sleep. All these affect the reliability and robustness of centralized WSNs negatively.

2.2.5 Low data privacy and security

Another drawback in large scale WSNs employing centralized data processing is the issue of security. The major problem with multi-hop transmission in centralized WSNs is that it is liable to attacks on the source data and nodes' identities during hopping. The reason is that for a resource-constraint WSN with source node sending data to the destination through several intermediary nodes, there is a possibility of intrusion, identity tracing by an adversary, gleaning, and modification of source data by the intermediary nodes [69]. Lastly, the lack of data privacy is another issue with centralized processing in WSN, as data is processed out of out the network (away from the sensor nodes) where it is collected.

Given that energy efficiency is a main requirement in WSN design and the need for a WSN to go unattended for a long period of time without replacement of the sensor nodes' energy sources, we will in the next section briefly discuss the techniques available in the literature for prolonging WSN lifetime.

2.3 Power management techniques for maximizing WSN lifetime

Energy efficiency is a critical aspect in most WSN applications because of the energy constrain of sensor nodes and the need to prolong WSN lifetime. As stated in the review of Yetgin et al. [70], there are different definitions of network lifetime depending on the specific application, the objective function and the network topology considered. However, we will stick to the definition of network lifetime being the total amount of time during which the WSN is capable of effectively performing its functions and meeting up with the application requirements. In that light, a WSN is considered dead when it is unable to forward any data to the base station [21].

As sensor nodes in most WSN applications are generally battery-powered, the WSN lifetime is constrained by the battery of the individual sensors in the WSN [16]. Thus, the critical issues to be considered when maximizing WSN lifetime are how to reduce the energy consumption of the nodes or how to replenish their energy sources in an efficient and realistic way [16], [71]. Most importantly, it is required that the WSN lifetime be long enough to permit the WSN fulfil the application requirements. The quest to prolong WSN lifetime, has in recent years led to a plethora of power management techniques in the literature. However, most of the techniques present in the literature can be broadly categorized into techniques either involved in reducing energy consumption of sensor nodes or those involved in replenishing the consumed energy in battery powered nodes [71].

In this section, we briefly discuss on WSN lifetime prolongation techniques available in the literature. We developed a taxonomy (Figure 2.2) for the classification of WSN lifetime prolongation techniques available in the literature. The prolongation techniques can broadly be classified as energy conservation techniques, energy balancing techniques, or energy harvesting techniques with each having a number of sub-categories as shown in Figure 2.2.

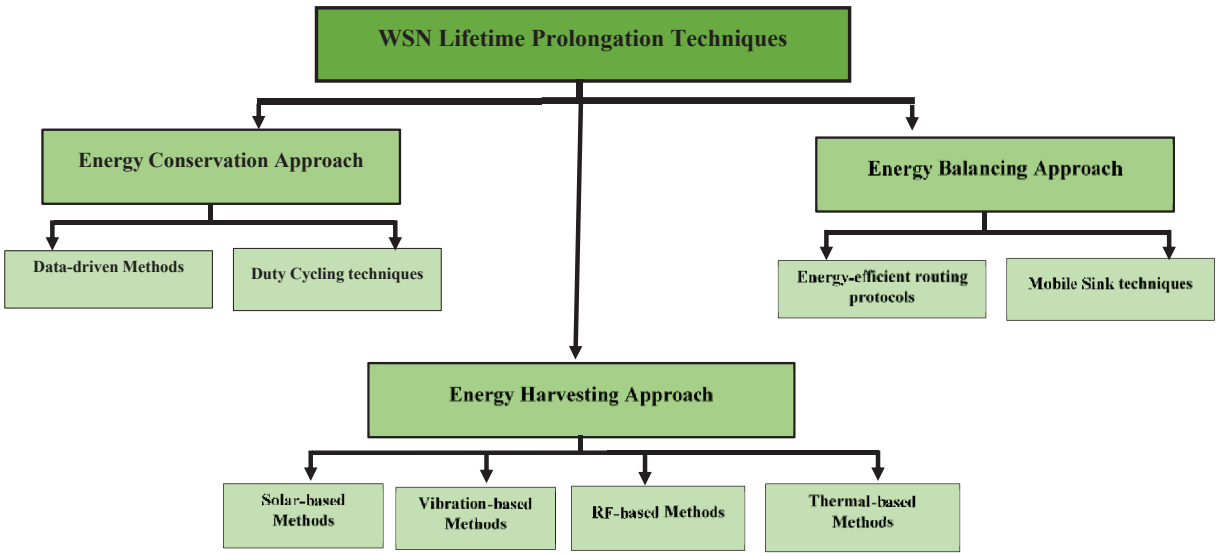

Figure 2. 2: Classification of WSN lifetime prolongation techniques

2.3.1 Energy harvesting techniques

Energy harvesting techniques convert energy from external sources (such as human and mechanical sources) or ambient environment sources (such as wind, vibrations, solar, acoustic, and thermal) into electrical energy that can be used to power wireless sensor nodes [71]. The energy harvested by these sources are used to replenish the energy depleted by the sensor node and thus prevent the development of an energy hole in the WSN. This increases the lifetime of the nodes and that of the WSN as a whole. Thus, the goal of energy harvesting techniques is to convert energy from one form to another that can be used to power sensor nodes and thus extend the lifetime of the WSN [72]. Moreover, as shown in Table 2.1, the source from which energy is harvested in a WSN is a valuable resource since it determines the amount of energy available to the network and the rate of conversion from the source to electrical energy [71]. This makes the ambient sources which are accessible within an environment and which do not need any external energy supply very attractive to WSN applications.

From the literature, the most common energy harvesting techniques include: solar-based, thermal-based, wind-based, vibration-based, and RF-based sources [71]–[73]. Prauzek et al. [72], reviewed and presented a comprehensive account of energy harvesting sources, energy storage devices, and

corresponding topologies of energy harvesting systems published within the last 10 years (from 2008 to 2018). In another study, Peruzzi and Pozzebon [73], in their review paper provided a detailed overview of the existing Low Power Wide Area Network (LPWAN) systems relying on energy harvesting for their powering. In [73], the different LPWAN technologies and protocols are discussed alongside the applicable energy harvesting techniques as well as presentations of the architecture of the power management units. Finally, the magnitude of the energy harvested from the different energy sources reported by [74], is shown in Table 2.1.

Table 2. 1: Amount of energy generated by different energy sources

Energy source	Typical power density	Embedded nominal power	Transducer
Solar	15 mW/cm^2	42 dBm (15 W)	Solar panels
Thermal	15 µW/cm^3	22 dBm (150 mW)	Thermoelectric Generator
Vibration	145 µW/cm^3	19 dBm (74 mW)	Electromagnetic
	330 µW/cm^3	–7 dBm (200 µW)	Piezoelectric materials
	50 µW/cm^3	–7 dBm (200 µW)	Electrostatic
Directed RF	50 mW/cm^2	20 dBm (100 mW)	Antenna
Ambient RF	12 mW/cm^2	–23 dBm (5 µW)	Antenna

2.3.2 Energy balancing techniques

Energy balancing techniques seek to ensure that the energy consumption is evenly distributed in the WSN such that nodes have a fairly equal amount of energy. This reduces the likelihood of a black hole (energy hole) developing in the WSN and prolongs the WSN lifetime [71]. Thus, the objective of the energy balancing techniques is to balance the communication burdens of the sensor nodes in the WSN by ensuring that they spend their energy at approximately the same rate. This is achieved via balancing the load of the sensor nodes in the WSN by employing techniques such as clustering. Clustering schemes are one of the most used methods for prolonging the network lifetime in WSNs via load balancing. They involve grouping sensor nodes into smaller groups called clusters, headed by a coordinator called the Cluster Head (CH). The CH performs specialized functions such as data fusion and aggregation, and communicates the aggregated data directly to the base station or to other CHs. The CH can be selected randomly or based on one or more criteria and this also largely affects the WSN lifetime. An ideal CH is the sensor node with the highest residual energy, the maximum number of neighbour nodes, and the smallest distance from base station [75]. The goal of clustering schemes is to reduce the number of redundant

communications in the WSN by reducing the number nodes that communicate with the base station. By performing aggregation on data within the cluster, the energy consumed in the network is far less than when all the raw data are sent to base station [40].

Other popular energy balancing techniques comprise on the one hand energy-efficient routing protocols with the ability to achieve uniform energy consumption and on the other hand mobile sink techniques that ensure that the energy consumed by nodes in a large-scale network is uniform by reducing the energy consumed by the sensor nodes when serving as relays. This is done by causing the sink to move closer to the sensor nodes to reduce the communication distance. Most of the energy-efficient routing protocols implement clustering since the cluster-based structure is considered by research community as an effective architecture for data-gathering in WSN [76].

2.3.2.1 Energy-efficient routing protocols

Energy-efficient routing protocols are needed for large-scale battery powered WSNs to ensure uniform energy consumption and load balancing. Besides, they are also needed to achieve reliable and real-time data forwarding to the sink. This has led to many research efforts devoted to the design of energy efficient routing protocols and/or enhancement of existing ones. Routing Protocol for Low-Power and Lossy Networks (RPL) and Low Energy Adaptive Clustering Hierarchy (LEACH) are among the well-known energy efficient protocols used in WSNs [77].

Heinzelman et al. [78] designed LEACH, a clustering-based routing protocol that minimises global energy usage by distributing the load to all the nodes at different points in time. The simulation results revealed that LEACH reduced the communication energy by 8 times when compared with direct transmission and minimum transmission-energy routing. In addition, the first node death in LEACH occurred 8 times later than the first node death in direct transmission, minimum-transmission-energy routing, and a static clustering protocol. Furthermore, the last node death in LEACH occurred over 3 times later than the last node death in the other protocols. The authors of [79] and [80] addressed the load imbalance problem and proposed new load balancing methods for the RPL protocol. Simulations were conducted in cooja simulator and the results revealed an improvement in load balancing, which resulted in a reduction in power consumption and an increase in network lifetime.

2.3.2.2 Mobile Sink techniques

In a typical large-scale WSN, the base station (sink) is static. As such, the data from the sensor nodes are transmitted to the base station through multi-hop communications. Hence, some sensor nodes in the WSN would not only sense and send their data, but also act as wireless relays that forwards the data of their neighbours towards the sink. Consequently, nodes near to the sink have their battery deplete faster, leading to nonuniform energy consumption which eventually causes the development of an energy hole in the WSN. The energy hole disables the WSN and thus reduces its lifetime regardless of the fact that there are still a number of sensor nodes in the WSN whose battery are not yet depleted. In recent years, contrary to static sink, the mobile sink approach has attracted much research interest because of increase in its potential WSN applications and its potential to improve network performance such as energy efficiency and throughput [76]. In this approach, the mobile sink visits the network in order to spread more uniformly the energy consumption [16]. However, this solution is not very common since the sink in most WSN applications is static.

Nasir et al. in [76] proposed a Mobile Sink based Routing Protocol (MSRP) for prolonging the lifetime of a WSN implementing the clustered architecture. In their proposed methodology, the mobile sink moves in the clustered WSN to collect sensed data from the CHs within its vicinity. The movement of the mobile sink is guided by the residual energy of the CHs thereby causing the mobile sink move to the CHs having higher energy. The high energy CHs then assume the responsibility of relaying the data transmitted by all the sensor nodes in the WSN via either single-hop or multi-hop communications. This results in the responsibility of relaying the data transmitted to the sink by far way sensor nodes alternating between different high energy CHs near the sink. The end result is the avoidance of an energy hole since there is a balance in the energy consumption and this improves the WSN lifetime. In a recent study, Zhang et al. [81] studied the lifetime of a mobile WSN where they used evolutionary computing algorithms to solve the mobile WSN model.

2.3.3 Energy conservation techniques

Conserving the energy of sensor nodes requires a compromise between various activities at both the node and network levels [21]. This can be done by implementing energy efficient protocols that are aimed at minimizing the energy consumption during network activities and/or

implementing power management schemes that involve switching off node components that are not temporarily needed. The reason is that a large amount of energy is consumed by node components (CPU, radio, sensor, etc.) even in the idle mode [16].

The techniques for minimizing energy consumption during network activities include energy-efficient routing protocols and implementation of mobile sinks (discussed in sub-section 2.3.2) while the techniques for minimizing energy consumption at the node level include duty cycling techniques (radio optimization, sleep/wake-up schemes, transmission power control, dynamic voltage frequency scaling), and data-driven approaches (data compression, data aggregation, data prediction, hierarchical sensing, adaptive sampling, and model-based active sensing) [16], [71], [77]. Figure 2.3 presents a commonly adopted taxonomy of energy conservation strategies used for preserving WSN lifetime.

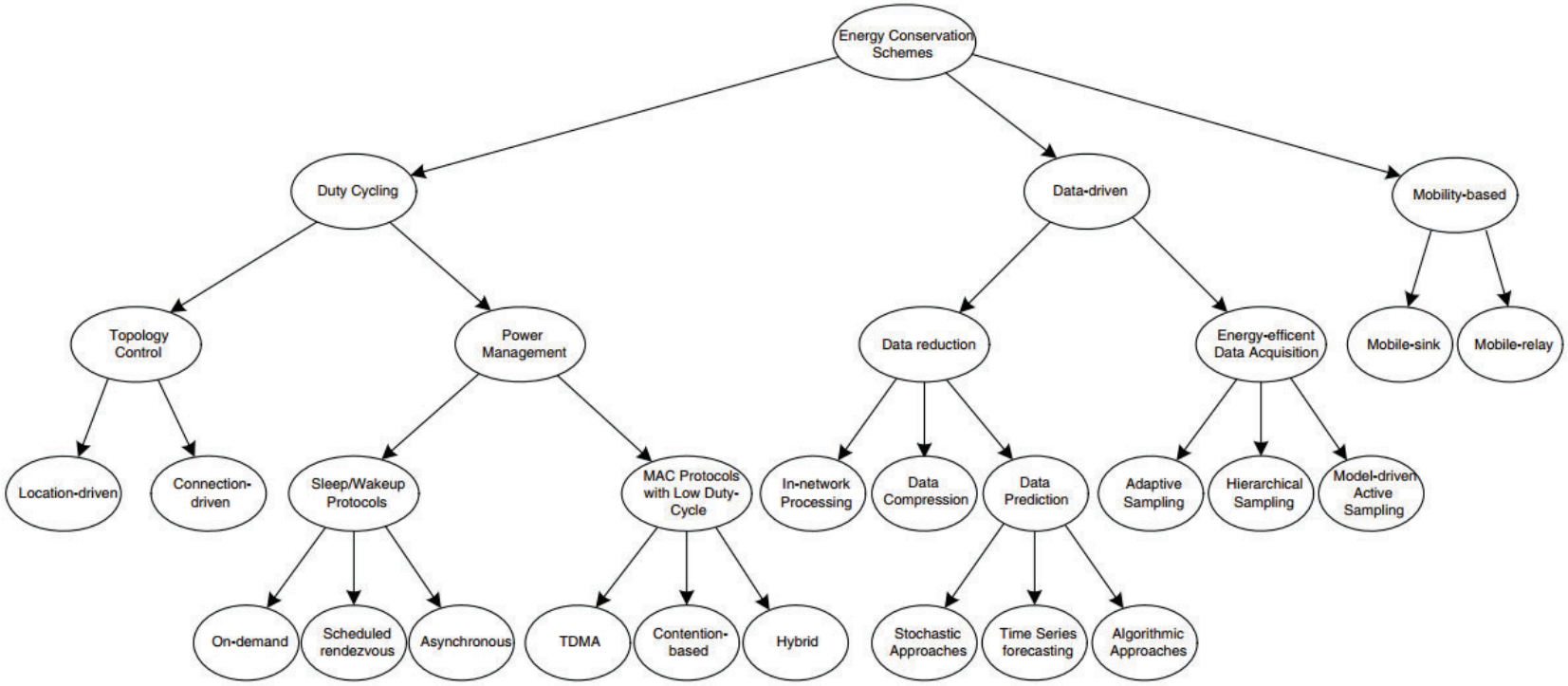

Figure 2. 3: Classification of energy conservation schemes for preserving WSN lifetime **[16]**

2.3.3.1 Duty Cycling Approaches

One commonly used method for reducing sensor node's energy consumption is by turning off sensor node's hardware components when they are not needed and waking them up whenever necessary. This establishes a small duty cycle for the nodes based on events occurring in the monitored environment [82]. Thus, techniques based on duty cycling rely on the fact that active nodes do not need to maintain their radios, processor, and sensing devices continuously on. According to the survey by Anastasi et al. [16], duty cycling is achieved by two complementary approaches, with one approach taking advantage of the redundancy in WSNs by adaptively selecting only a minimum subset of nodes to remain active for maintaining connectivity while the other approach ensures that active nodes do not maintain their radio continuously on by constantly switching them off (i.e., put it in the low-power sleep mode) when there is no network activity. The authors termed the former topology control and the latter power management.

As generally known in the literature, the communication unit (radio module) consumes the most part of the sensor node's energy [1], [17]. By reducing the activities (transmission, reception, idle listening) of the radio module, higher energy savings can be achieved at the node level. This is because the energy consumption of the radio module is of the same order of magnitude in the reception, transmission, and idle states, while the power consumption drops significantly in the sleep state [16], [17]. Therefore, the radio should be put to sleep (or turned off) whenever possible. Additionally, there are several factors that affect the power consumption characteristics of a radio module, including the radio duty cycle, modulation scheme, data rate, and transmission distance. To optimize the radio and minimize sensor node energy consumption, techniques used include sleep/wake-up schemes and optimization of radio parameters such as radio coding and modulation techniques, transmission power, and antenna direction [71]. Sleep/wake-up schemes adapt the node to the activities of the network to conserve energy by putting the radio to sleep (to minimize idle listening) while the transmission power control approaches dynamically adjust the transmission power of the radio to maintain an effective communication link between pairs of nodes while not transmitting at full power capacity. Furthermore, dynamic voltage frequency scaling is another means of achieving energy savings at the node level. This technique harnesses the ability of modern microprocessors to operate at different voltage and frequency settings in the

active state to achieve additional energy savings [77]. It also optimizes the sensor node's performance in the active state by adjusting the voltage and the operating frequency of its CPU based on the instantaneous computational load requested by the application [82].

2.3.3.2 Data-driven Approaches

Data-driven approaches can be classified into either data reduction or energy-efficient data acquisition schemes depending the problem they address [16]. The data reduction schemes on the one hand are primarily involved in reducing the number of data transmissions and the amount of data transmitted, as data moves from the sensor nodes to the base station. Energy-efficient data acquisition schemes on the other hand are involved in reducing the energy spent by the sensing subsystem by typically reducing the number of samples generated by the sensors.

The data reduction schemes include in-network processing techniques such as data aggregation, data compression, and data prediction [16], [71]. Data aggregation techniques increase the network lifetime by fusing data in an efficient manner as it traverses the network from one node to the another until it gets to the sink [83]. Likewise, data compression involves encoding information at the sensor nodes and decoding it at the sink and it can be applied to reduce the amount of information sent by source nodes [16]. This reduction in the amount of data transmitted and received also reduces the radio module's active time, which also decreases the sensor node's energy consumption. Similarly, data prediction techniques have as aim to reduce the network power consumption by minimizing the communication cost. They achieve this by building an abstraction of a sensed phenomenon, i.e., a model describing data evolution. The model can predict the values sensed by sensor nodes within certain error bounds, and resides both at the sensors and at the sink [16]. Transmissions between the nodes and the sink occur only when sensor nodes measure values outside the threshold of the prediction models [71]. This reduces the frequency of transmission and the energy needed for communication as well [16], [71]. Anastasi et al. [16] categorized data prediction techniques into: stochastic approaches, time series forecasting, and algorithmic approaches. Stochastic approaches such as Kalman Filter (KF) first derive a state space representation of the monitored phenomenon, then future samples can be guessed by filtering out a non-predictable component modelled as noise [59]. Time series forecasting techniques such as Moving Average (MA), Auto-Regressive (AR) or Autoregressive Moving Average (ARMA)

models, etc. are involved with predicting future values from historical values (time series) derived from periodic sampling. Lastly, algorithmic approaches rely on a heuristic or a state-transition model describing the sensed phenomenon.

The energy-efficient data acquisition schemes include adaptive sensing techniques such as hierarchical sensing, adaptive sampling, and model-based active sensing [82]. By reducing the number of samples generated by the sensors, an efficient sensing strategy also reduces the amount of data to be processed and possibly transmitted by sensor nodes, and thus generate further energy savings [82]. In hierarchical sensing, a sensor node has multiple sensing devices monitoring the same physical parameter, but with each having a different sensing accuracy and power consumption. Accuracy can be traded-off for energy efficiency by using the low-power sensors to get a rough estimate of the monitored parameter [16]. Once an event has been detected, the accurate power hungry sensors can be activated to give more accurate readings of the physical property at the cost of greater energy consumption [82]. Adaptive sampling strategies reduce the number of measurements and the communications required to achieve an accurate estimate by exploiting the spatio-temporal correlations between the sensed data. This is achieved by first reducing the number of active sensor and also dynamically adjusting the sampling rate. They reduce the number of samples by exploiting spatio-temporal correlations between data [16], which in turn reduces the energy spent on sensing. Spatial correlation on the one hand tries to reduce the sensing energy consumption by exploring the fact that measurements taken by sensor nodes that are spatially close to each other do not differ significantly. In this light, only a few sensors can be activated. On the other hand, temporal correlations are based on the idea that if the monitored phenomenon does not change rapidly, then the sampling rate can be reduced without any loss of relevant information [82]. Finally, model-based active sampling reduces the number of data samples by using a computational model [16]. This involves the use of forecasting models to build an abstraction of the sensed phenomenon. The forecasting model is built with an initial set of sampled data. Then, the model is used to predict the data instead of performing a continuous sampling in the field [59]. Therefore, the energy dissipated for data sensing and transmission is saved. [82].

As has been shown in the presented literature, the data reduction techniques (which are applications of distributed computing in WSN) can be used to extend WSN lifetime. We saw that

distributed computing has the potential to reduce sensor nodes' energy consumption and extend the WSN lifetime by using the on-board computing resources embedded within the sensor node to reduce the number of messages and the amount of data transmitted. This implies that the data processing task will be performed using the on-board processing unit of the sensor nodes and only decisions will be sent to the base station. Since this thesis is focused on distributed computing in WSN and how it can be used to extend WSN lifetime and increase performance, we will in the next section discuss the advantages of distributed processing in WSN especially in prolonging WSN lifetime and providing the possibility of real-time monitoring in WSNs.

2.4 Distributed Computing in Wireless Sensor Networks

In this section, we provide a brief review of distributed computing in WSNs by looking at the motivation for distributed computing in WSNs and its benefits. We also survey some studies that have applied distributed computing in WSNs.

2.4.1 The relevance of distributed computing in WSN

Based on the drawbacks of the centralized computing mentioned in section 2.2, a logical thing to do is to perform distributed computing within the WSN [17]. The core idea of distributed computing in WSN is to invest more into computation within the network by harnessing the onboard processing capabilities embedded in each node for local processing whenever possible to reduce communication costs. In this distributed approach, each node performs processing on its local data and only communicates with its direct neighbours to reach some desired accuracy. This has the potential to significantly reduce the amount of multi-hop communications and also eliminate data processing at the base station, since only final decisions will be sent to the base station. Such a system will be suitable for real-time monitoring of autonomous systems where there is need to prioritize local and real-time decisions [67].

In recent years, advances in microelectronics has resulted in the development of powerful and low-power processing units (e.g. microcontrollers) with higher computational power and memory capacity [4]–[6], [84]. This has also reduced the challenges of embedding intelligent data processing on sensor nodes, causing a paradigm shift where distributed computing is increasingly becoming very popular in most WSN monitoring applications [4], [5], [84]. The presence of

distributed sensing in a WSN and the availability of sufficient computing resources at each sensor node can be properly harnessed by using distributed algorithms that minimize communication and energy costs, as well as provide robustness to node failures [19]. This creates a scenario where the sensor nodes can communicate among themselves and perform distributed computation over the sensed data to identify the occurrence of an event [8]. This improves on the scalability of WSNs, reduces latency as well as network energy consumption, and also improves data security and privacy [13], [15], [18], [67].

2.4.2 WSN and Edge Computing

In the last decade, the term Internet of Things (IoT), has progressively gained dominance as the keyword to define connected embedded devices. It replaces the pioneer term WSN, which is one of the first physical implementations of Ubiquitous Computing, and finally integrates it as a part of IoT (the physical network mainly used for monitoring).

With research progress in this field, numerous computing paradigms have emerged such as Mobile Cloud Computing (MCC), cloudlet computing, mobile clouds, mobile IoT computing, IoT cloud computing, fog computing, Mobile Edge Computing (MEC), and edge computing [85]. The latter, edge computing [86], could be considered as a related field of WSN distributed computing. However, what differs between WSN and IoT is the hardware computation capacity. Recently, the Internet Engineering Task Force (IETF) [87] standardized a classification for hardware devices used in IoT by demonstrating that a performance gap exists between the lowest class, i.e., Class 0, used in WSN [88] and Class 1 and above, which comprise of hardware devices commonly found in IoT [89].

Furthermore, given that wireless sensor nodes are highly constrained in terms of computation capacity and energy consumption, the promising capacity of edge computing has not been evaluated in this thesis due to the gap that exist between the hardware capacity of the hardware devices involved in each of the networks (WSN and IoT) [90] and those developed using a state-of-the-art energy-aware hardware design for edge computing by Jiang et al. [91]. Even though paradigms and algorithmic propositions (such as deployment of artificial intelligence at the edge) emerging from edge computing could be of interest, issues such as portability efforts and shrinking

requirements needed to transfer software from medium- to high-performance computing units to highly energy-constrained and low-power computation capacity hardware targets, limited this thesis to focus only on distributed computing for low-cost/low-performance, battery-powered, and highly constrained sensor networks [59].

2.4.3 A Survey on Distributed Computing in WSN

According to Stula et al. [92], distributed computing is any process conducted by multiple agents or entities that perform operations on information and together generate resulting information, and can be defined and observed in terms of memory, communication, and processing. Huang, in [9], classified the applications of distributed computing in WSN according to the following taxonomy: Distributed Query Processing, Collaborative Signal Processing, or Distributed Estimation and Detection. Additionally, other applications of distributed computing in WSN include: local (in-node or on-sensor or edge) processing and in-network processing.

2.4.3.1 In-network Processing

In-network processing involves the processing of data as it travels via the WSN to the sink. It involves actions such as fusion and aggregation on the data as it moves within the WSN from one sensor node to another. This reduces the number of redundant data that needs to transmitted. For example, Serpen and Liu [20] demonstrated through simulations, a case study that leverages existing WSNs as a parallel and distributed hardware platform to implement computations for artificial neural network algorithms. The results of their simulation suggested that the WSN-based neurocomputing architecture is a feasible alternative for realizing parallel and distributed computation of artificial neural network algorithms. However, their study did not consider the energy constraint of WSN as they assumed that the sensor nodes were not limited in energy supply. In another study, Pascale et al. [18] proposed an in-network processing framework to tap into the collective computation capability of WSN devices by coupling data communication and processing for the transformation of raw data into appropriate actions as it travels via the network towards the actuating nodes. Their results showed that distributed computing (via in-network processing) decreases latency and improves the balancing of energy consumption among the sensor nodes, thus mitigating the energy hole effect and increasing the expected lifetime of the network. However, the results were validated by simulations in Cooja.

2.4.3.2 Collaborative Signal Processing

In collaborative signal processing, also referred to as Wireless Distributed Computing (WDC), a master node which needs to perform a complex computational task in a limited time frame divides the computational task into a number of subtasks and then assigns these subtasks to some slave nodes (neighbouring nodes) [93]. In [14], the authors discussed some of the possible applications of WDC such as image processing and pattern recognition, distributed data storage and database search, Synthetic Aperture Radar (SAR) processing, etc. Energy savings in WDC were demonstrated in this study by a wireless ad-hoc network comprised of a tactical handheld, radio nodes attached to a UAV, or sensor nodes. The findings showed that the reduction in energy consumption of the wireless nodes was achieved firstly by the fact that WDC enables processing within the network which reduces number of bits transmitted over the long backhaul at the cost of computational energy consumption. Chiasserini [94], extended the concept of collaborative signal processing in WSN by using a collaborative computational algorithm and communication scheme where the sensor nodes were made to operate as a Distributed Digital Signal Processor (DDSP). Fast Fourier Transform (FFT) algorithm was applied to the DDSP approach and the results showed that the energy consumption obtained at different processing frequencies when the FFT is computed by a single sensor node were higher compared to the results derived in the case where the computation is distributed among multiple sensor nodes.

2.4.3.3 Local Processing

In the case of local processing, raw data are processed locally at the sensor node using its processing unit, and only the analysed results are transmitted via the WSN to the sink. Data processing is done independently on the sensor nodes and the sensor nodes do not collaborate with other [65]. Feng et al. [95], implemented an envelope analysis algorithm on a WSN node composed of a cortex-M4F core processor for feature fault extraction in a condition monitoring application. The results from the study showed that the sensor node was able to identify simulated faults and achieve real-time condition monitoring while reducing the data transmission throughput by 95%. In another study, Kartakis et al. [13], presented an end-to-end water leak localization system, which exploits edge processing in battery-powered sensor nodes. The sensor nodes were based on Intel Edison development boards and the proposed system combined a lightweight edge anomaly

detection algorithm based on a Kalman filter and compression rates and a localization algorithm based on graph theory. According to the authors, the edge anomaly detection and localization elements of the system produced a timely and accurate localization result which reduced the communication by 99% compared to the traditional periodic communication.

2.4.3.4 Distributed State Estimation and Event Detection

In distributed state estimation and detection applications, the WSN makes a decision about the value of a physical variable (estimation) or the occurrence of an event (detection) in a distributed manner [9]. Distributed State Estimation (DSE) algorithms implement distributed data fusion, where neighbouring nodes communicate with each other to improve the accuracy of the monitored parameter whereas Distributed Event Detection (DED) enables evaluation of gathered data in a cooperative way within the network to detect the occurrence of an event [96]. In the DSE study by Alriksson and Rantzer [97], the performance of a local Kalman filter and a distributed Kalman filter was evaluated experimentally using an ultrasound-based positioning application composed of a sensor network with seven sensor nodes. There was no centralized computation and the goal was to make sure that every node in the network has an accurate estimate by performing computations in a distributed manner and communicating only once per sampling interval. Dziengel et al. [96], presented the deployment and evaluation of a fully applicable DED system for fence monitoring. Their goal was to solve the classification problem of distinguishing trained events within a WSN at a construction site fence and for which they implemented an in-network computation solution that uses data compression methods based on a classical pattern recognition system. Their solution deployed a WSN composed of 49 nodes which were integrated in the construction site fence elements and they evaluated the acceleration data of all triggered nodes for 10 event classes within the network. Their results revealed that the in-network processing capability of their proposed solution clearly reduced the energy consumption beyond a communication distance of two hops to the base station and led to a prolonged lifetime of the network, while achieving an average event detection accuracy of more than 93%. In another study, Titouna et al. in [98] presented a Distributed Outlier Detection System (DODS) that implemented in-network outlier detection in which multiple sensed data types were considered and where outliers were detected locally by each node using a set of classifiers, so that neither information

about neighbours is needed to be known by other nodes nor a communication is required among them. The functionalities of the proposed scheme were validated via extensive simulations using real sensed data obtained from Intel-Berkeley Research Lab. The results showed that the proposed solution had an interesting performance and also consumed less energy.

Now that we have established the importance of distributed computing in WSN and given that WWPM can serve as a good application area for distributed computing in WSN, we will in the following sections present the state-of-the-art of WWPM. In addition, since this thesis is interested in improving the performance and extending the lifetime of WWPM systems, we will review a number of signal processing techniques used for improving the performance (accuracy) of leak detection and the power management techniques existing in the literature for the extending the lifespan of WWPM systems.

2.5 State-of-the-art of Leak Detection Techniques

2.5.1 Taxonomy for Classifying Leak Detection Techniques

In this sub-section, we present a general classification of the leak detection techniques used in water pipeline monitoring (WPM) and then focus on WWPM in the next section. Several taxonomies exist in the literature for classifying leak detection techniques, and there is no single unified taxonomy generally accepted. However, we developed a taxonomy for the classification of leak detection techniques, as shown in Figure 2.4. Based on our taxonomy, leak detection techniques used for WPM can be categorized as either static or dynamic depending on the mobility of the equipment used for detecting the leak signals. In addition, depending on whether the leak detection technique monitors external or internal parameters of the pipeline and depending on the usage of either specialized hardware equipment or low-cost sensors with computational algorithms for processing the leak signals, leak detection techniques can also be classified as external (hardware-based) or internal (software-based) methods. The software-based methods can either employ WSNs for monitoring the pipeline, in which case they are referred as WWPM techniques or use non-WSN technologies (e.g., wired telemetry and Supervisory Control and Data Acquisition –SCADA, that makes use of sensors connected to the main control centre via communication means such as copper cables or optical fibres) in which case they are referred to as non-WSN-based techniques. The WWPM techniques can be further categorized into either intrusive methods

that make use of invasive sensors such as pressure, flow sensors, etc. to monitor internal pipeline parameters or non-intrusive methods that make use of non-invasive sensors such as accelerometers, acoustic sensors, etc. to monitor external pipeline parameters such as pipe surface acceleration, sound of leak signals, etc. Finally, depending on the data processing technique used for analysing leak signals, leak detection techniques for WPM can be classified into either model-based, transient-based, signal processing or data-driven methods.

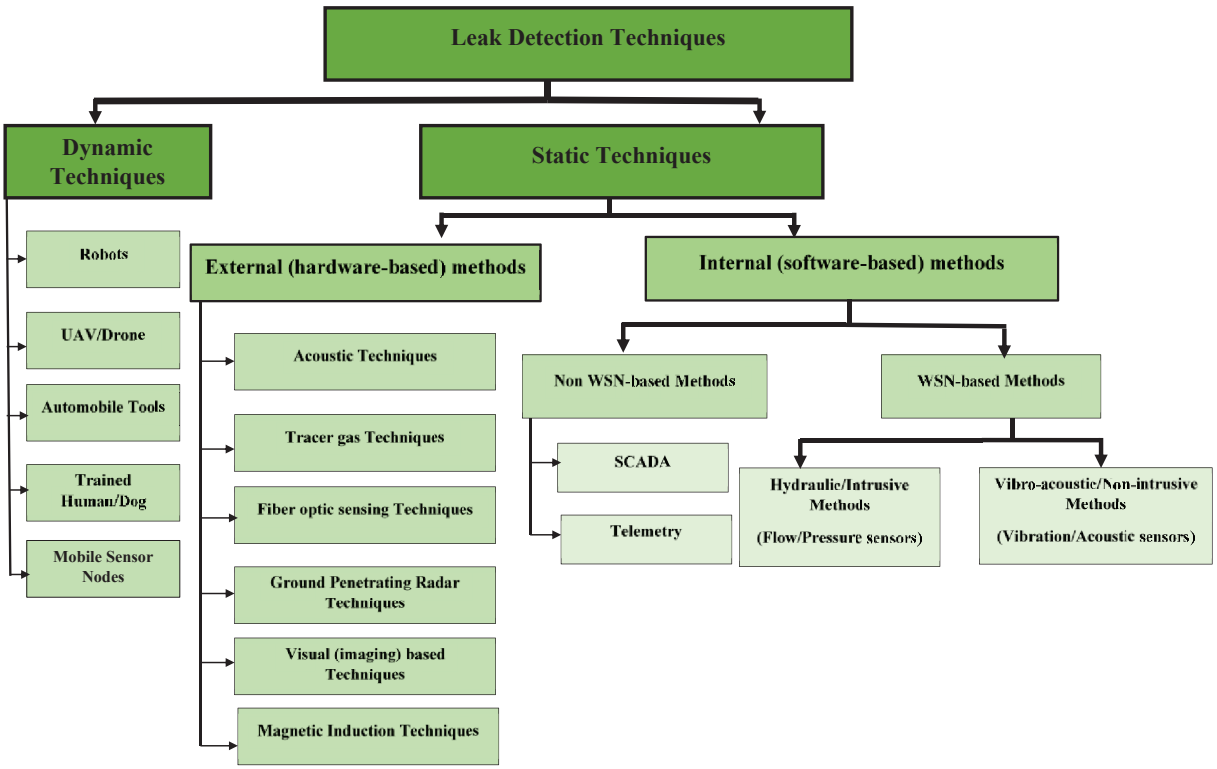

Figure 2. 4: Taxonomy for the classification of leak detection techniques.

2.5.2 Classification of Leak Detection Techniques

The detection and localization of water pipeline leakages are important to water utility companies because of the need to conserve raw/treated water and save associated costs [26]. A lot of research efforts have been dedicated to the development of a vast variety of techniques for leak detection and localization to minimize water losses caused by leaks [36]. Based on their technical approach, Adedeji et al. [27], Baroudi et al. [57], Adegboye et al. [99], and Torres et al. [44] in their survey papers on pipeline monitoring broadly categorized leak detection techniques as either external or

internal. In other surveys, Ismail et al. [37] classified leak detection techniques into software-based methods and hardware-based methods, Chan et al. [100] classified them as active and passive systems, while Zaman et al. [47] classified them into direct and indirect methods. The software-based and hardware-based methods of [37], active and passive systems of [100], and direct and indirect methods of [47] are likened to the internal and external methods of [27], [44], [57], [99], respectively, as shown in Figure 2.4. In another study, El-Zahab et al. [36] classified leak detection systems into two major classes, i.e., static leak detection systems and dynamic leak detection systems. Static leak detection systems rely on sensors capable of sensing leak signals and coupled with a communication technology while dynamic leak detection systems require the mobilization of a leak inspection team that carry specialized devices to the suspected leak site to perform an inspection and confirm or clear the suspicion [36]. Most software-based (internal or active) and hardware-based (external or passive) leak detection techniques are static, as shown in Figure 2.4.

In the sub-sections below, we briefly explain software-based and hardware-based methods of leak detection, highlighting the various techniques in each of the categories and stating their advantages and disadvantages.

2.5.2.1 Software-based Methods

The software-based methods use field sensors to monitor the operational and hydraulic conditions of the pipeline, such as the measurement of the flow, pressure and temperature [10], [37], [57], [99] and smart computational algorithms to process the measurements in order to detect and localize the occurrence of leaks on the pipeline [44]. Some of the software-based methods available in the literature include balance/model-based methods (e.g., Kalman Filter [26], [44], [101], State Observer [102], System Identification [103], Impedance Method [104], Mass Balance [105], Real-Time Transient Model [106]), pressure/transient-based methods (Negative Wave Pressure [40], Pressure Point Analysis [107], Inverse Transient Analysis [108], Transient Reflection Method [109], Transient Damping Method [110]), methods based on signal processing (Fast Fourier Transform [111], Wavelet Transform [112], Kalman Filter [13], [38], [50], Acoustic Correlation Analysis [113], Spectral Analysis Response [54]) and data-driven methods (Support Vector Machine [114], K-Nearest Neighbour [115], Naïve Bayes [114], Artificial Neural Network [116], [117]). They involve the use of either intrusive sensors or non- intrusive sensors to monitor the

internal pipeline parameters. As an advantage, they can make use of a WSN and they are also cost effective. We classified the software-based leak detection techniques that incorporate WSN as shown in Figure 2.5.

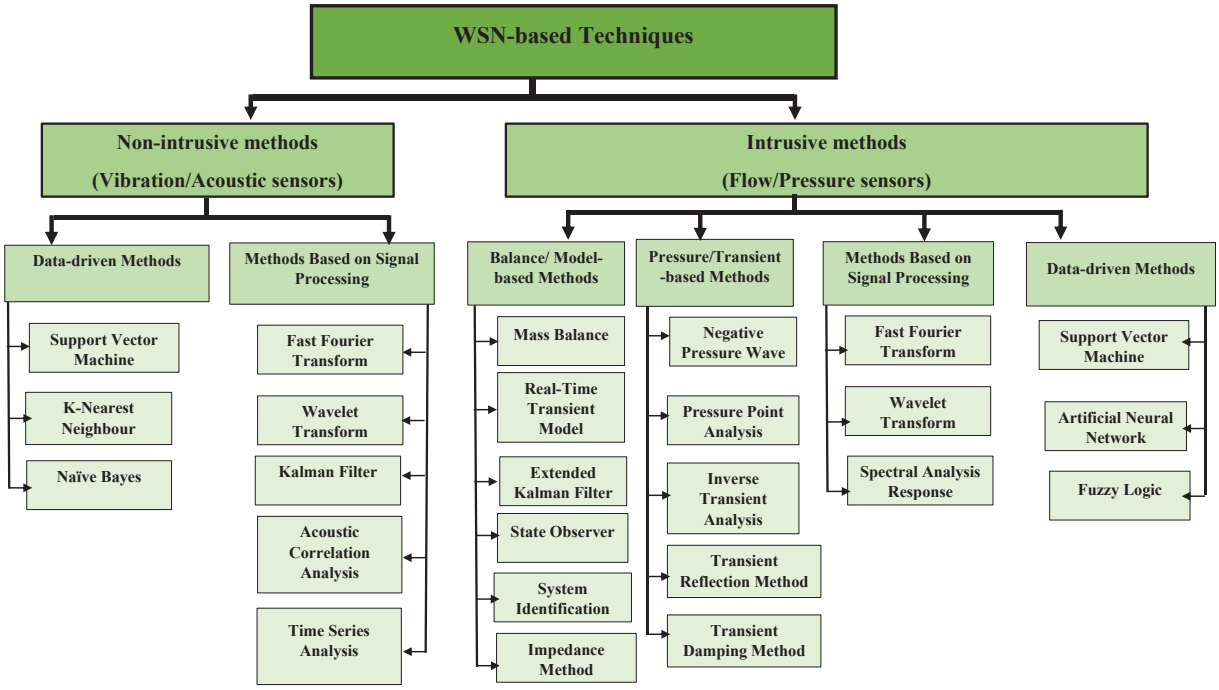

Figure 2. 5: WSN-based leak detection techniques

According to Figure 2.5, the main computational algorithms used for processing and analysing leak signals from field sensors to detect the presence of leaks on a WDN can be categorized into signal processing, model-based and data-driven algorithms. The signal processing algorithms extract information from the measured data and compare it with data sets from a fault-free benchmark to detect the presence or absence of a leak. Most signal processing algorithms analyse data in the frequency domain, and thus require some mathematical conversion [118]. This makes them computationally intensive and will result in huge power consumption when implemented on sensor nodes. However, the main advantage of this method is that an accurate mathematical model of the pipeline is not needed [119]. The model-based methods usually involve the use of mathematical functions or formulas to represent or replicate the operation of a WDN. They can determine the approximate leakage location by comparing pressure or flow measurement with the estimate obtained using the hydraulic network model [100]. The drawback with the model-based

methods is that they require a precise mathematical model of the pipeline system in order to accurately detect leaks. With the data-driven methods, large amounts of data are being collected and used to analyse, interpret, and extract useful information for operational and other purposes based on machine learning techniques and other data-driven methods [35], [101]. Their drawback is that they need a large amount of data and a long training time [101].

2.5.2.2 Hardware-based Methods

The hardware-based methods detect the presence of leaks from outside the pipeline by visual observation or by using specialized equipment that range from simple listening rods to more sophisticated approaches such as inspection gauges sending magnetic fields, electromagnetic waves or ultrasound through a pipeline's walls [10], [37], [57], [99], for physical monitoring. They use local sensors to send an alarm when a leak occurs, and do not perform computation for diagnosing a leak [44]. Some examples available in the literature include acoustic techniques [120], tracer gas techniques, fibre optic sensing techniques [121], ground-penetrating radar techniques [122], Magnetic induction techniques [123], etc. As a disadvantage, they may involve the use of expensive instruments, some are labour intensive and also do not make use of WSNs in monitoring. As an advantage, they are highly sensitive to leaks.

Figure 2.4 illustrated the classification of leak techniques into static and dynamic methods, followed by hardware-based and software-based methods while Figure 2.5 depicted the detailed classification of WSN-based leak detection techniques. A detailed review of these techniques along with a comparison can be found in [10], [27], [34], [37], [47], [57], [99], [100].

It should be noted that WPM schemes that incorporate WSNs are advantageous because WSNs provide effective solutions for pipeline monitoring, due to their low-cost, flexibility and ease of deployment in inaccessible terrain [10], [46], [57]. Consequently, we will focus our study on leak detection in WWPM systems. We will in the next section review some works in the literature that used WWPM systems to perform leak detection in WDNs.

2.6 Wireless Sensor Network Based Water Pipeline Monitoring

2.6.1 Introduction

WWPM systems are practically developed from two main parts: the sensors installed along the pipeline that periodically collect useful information relating to some pipeline parameters and the algorithms that process this information in order to detect and localize leaks in the pipeline [38]. Typically, the remote field sensors provide data to a centralized monitoring station (for centralized systems) or fusion centre (for decentralized systems), where the data undergo processing and are later fed into a leak detection algorithm to determine the presence or absence of a leak.

The sensors used for collecting measurements from the pipe in WWPM systems can either be intrusive or nonintrusive sensors. On the one hand, the intrusive techniques (i.e., destructive methods which puncture the pipes to place sensors such as flowrate metres and/or pressure gauge) are expensive, consume more power, and difficult to install, but however provide a higher accuracy. On the other hand, nonintrusive techniques (i.e., non-destructive methods that use sensors such as accelerometers, acoustic sensors, force sensitive resistors, vibration sensors, etc., installed on the pipe surface) are low-cost, consume less power, easy to install, but however provide low leak detection accuracy. In spite of the low detection accuracy of non-intrusive sensors, the requirement for WWPM systems to be low-power and cost-effective, has made the use of low-cost nonintrusive sensors in the design, implementation, and deployment of WWPM systems very popular in recent years [35], [43], [45], [46], [58]. Consequently, we will focus our study on WWPM systems that make use of low-cost nonintrusive sensors and we will review some works on WWPM that employ low-cost nonintrusive sensors for monitoring.

Nevertheless, we cannot proceed without mentioning that one of the biggest challenges with leak detection in WWPM using low-cost nonintrusive sensors is that the measurements collected by these sensors may be inaccurate due to their low sensitivity and the presence of measurement noise that may result in false alarms in the leak detection system. Therefore, the issue of reliably identifying a leak event in the midst of errors from a number of sources (commonly called noise) is a fundamental challenge of any leak detection system using low-cost sensors [46], [48]. However, to increase the leak detection accuracy, most WWPM solutions use computational

algorithms to increase the accuracy of the data measured by the low-cost sensors attached to the pipeline.

In sub-section 2.6.2 we will detailly discuss the challenges of using nonintrusive sensor for leak detection in plastic pipes, in sub-section 2.6.3 we will present some computational algorithms based on signal processing techniques that have been used for leak detection, in sub-section 2.6.4 we will review some selected vibration-based WWPM studies, and finally in sub-section 2.6.5 we will review studies that have applied one or more power management techniques, which we presented in section 2.3, for extending the lifespan of their proposed WWPM solution.

2.6.2 Challenges of Leak Detection in Plastic Pipes using Nonintrusive Sensors

The occurrence of a leak on a pipe generates noise (a leak signal) that propagates on both sides of the leak position. The noise propagates along the pipe, both in the fluid (which can be detected by hydrophones) and along the pipe-wall (which can be detected by accelerometers). Appropriate sensors placed at convenient locations on both side of the leak position can be used to detect these leak signals [124] as shown in Figure 2.6.

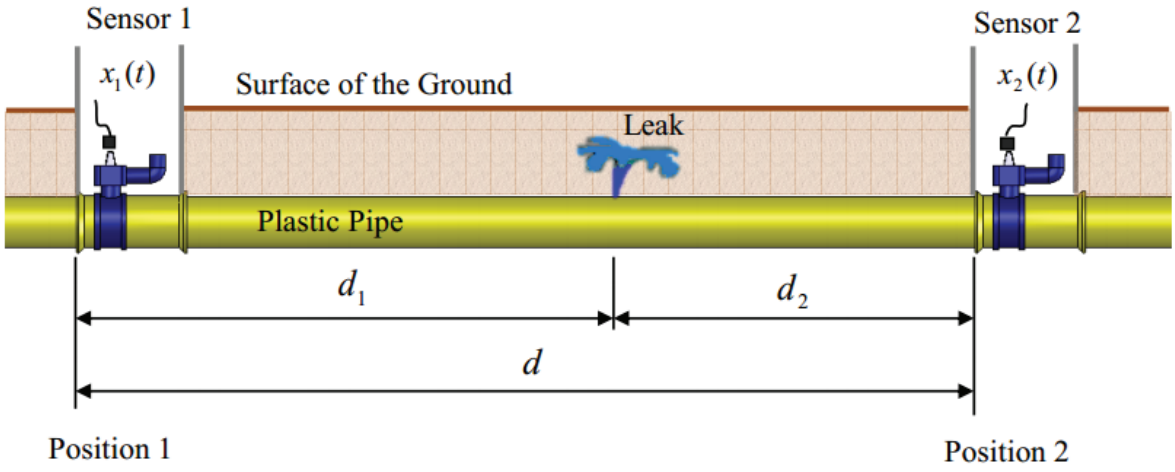

Figure 2. 6: Leak detection by acoustic/vibration sensors **[125]**

The reliable detection of leak signals in WDN using non-intrusive sensors such as accelerometers is dependent on a number of factors such as pipe size and type; leak type and size; pipe pressure; interfering noise; and the sensitivity and frequency response of the accelerometer [126]. The pipe

material and diameter have the most significant effect on the attenuation of leak signals in the pipe. From the literature, it is known that leak signals travel farthest in metal pipes and are attenuated greatly in plastic pipes [55]. The attenuation is even greater for larger diameter pipes [126], thus making it harder to detect leaks. Furthermore, the pipe material and diameter also affect the predominant frequencies of the leak signals. The larger the diameter and less rigid the pipe material, the lower the predominant frequencies. This effect makes leak signals susceptible to interference from low-frequency vibrations, such as vibrations from pumps and road traffic [126]. In addition, the vibration is also proportional to the pressure, with the leak induced energy intensity being greater for higher pressures.

Accelerometers respond to acceleration and so tend to be more responsive to higher frequencies. They are most effective on metallic pipes and tend to be less effective with non-metallic pipes. This is due to the increased attenuation of the high frequency components of the leak signal as they propagate through the pipe. This attenuation is partly caused by damping in the pipe wall and radiation of noise into the surrounding medium [125]. To reliably detect leaks in metallic pipes using accelerometers, the inter-sensor spacing can be as large as 500 m, even though a maximum spacing of 200 m is recommended [46]. However, for plastic pipes, the inter-sensor spacing cannot exceed 100 m (even in the case when high accuracy accelerometers such as those of the B&K brands are used). Thus, for reliable leak detection over longer distances in plastic pipes, the use of hydrophones is recommended. However, hydrophones are intrusive (destructive) methods of leak detection that require the sensor to be placed into the water at convenient fittings so as to detect the leak signal. This makes them difficult to install and not attractive in our context. Despite the drawbacks of accelerometers in detecting leaks in plastic pipes, their low-cost (especially MEMS accelerometers) and their ease of installation makes them very attractive for usage especially in our context.

Conclusively, the challenges associated with leak detection in plastic pipes using low-cost MEMS accelerometers can summarised under:

1. Short inter-sensor distance
2. High false alarm rate

The short inter-sensor distance is due to the higher wave attenuation in plastic pipes causing leak signals not to go far [54]. This will require the sensors to be placed very close to each other to have a higher spatial resolution to permit them reliably detect leaks. Using high-cost sensors such as B&K accelerometers will result to an increase in the cost of the WWPM system. However, a reduction in the cost of the WWPM system can be achieved via the use of low-cost MEMS accelerometers, as demonstrated in the works of Ismail et al. [58]. Unfortunately, low-cost sensors have lower accuracy and this leads to an increase in the false alarm rate of the leak detection system.

The high false alarm rate is due low sensitivity of low-cost MEMS and the difficulty in reliably capturing leak signals in the presence of environmental or background noise (such as vibrations from opening/closing the taps at user premises, moving vehicles, starting/stopping of pumps, opening/closing of valves, etc.). This results in false alarms caused by this environmental perturbations unrelated to the state of the pipe [46], [127].

Decreasing the cost of the leak detection system by using low-cost MEMS accelerometers for leak detection in plastic pipes will eventually decrease the performance (accuracy) of the leak detection system. However, the accuracy of leak detection systems using low-cost MEMS accelerometers can be improved by employing some signal processing techniques [35]. The major role of the signal processing techniques is to extract features related to leak in the pipe and to further process them to improve on the leak detection accuracy.

In the next sub-section, we discuss different signal processing techniques that have been used for leak detection in plastic WDNs using WWPM systems that employ nonintrusive sensors for monitoring.

2.6.3 Signal Processing Techniques used for Leak Detection in WWPM

Generally, signal processing methods are involved with extracting certain features from the captured signal. Subsequently, the extracted information, such as amplitudes, wavelet transform coefficients and other frequency response components are used to determine leak events by comparing them with a known threshold value that corresponds to a no-leak scenario [27], [51], [99]. Adegboye et al. [99], summarized the signal processing approach for leak detection in five

steps which include: (1) Data acquisition which involves the measurement of leak signals by field sensors (2) Signal pre-processing which involves pre-processing measurements to filter the background noise; (3) Feature extraction which involves applying various statistical, spectral and signal transform techniques to extract relevant features used to monitor the state of the pipeline; (4) Pattern classification which involves making a decision by comparing the pattern of the extracted feature with that of known pre-set signal or previous features; (5) Leak detection which is achieved by comparing the extracted pattern with the threshold value. Figure 2.7 depicts the step-by-step procedure of leak detection in WDN using signal processing techniques.

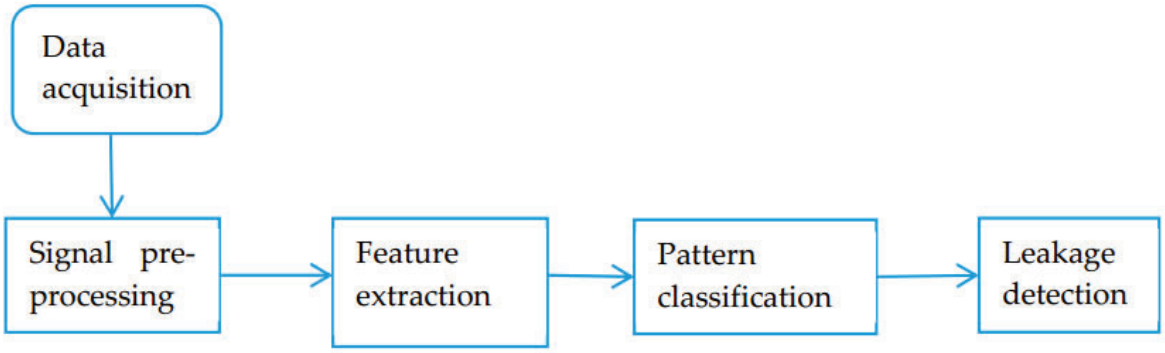

Figure 2. 7: Step-by-step procedure for leakage detection in WDN using Signal processing techniques **[99]**

The signal processing algorithms used for leak detection in WWPM can be broadly categorized into algorithms that operate entirely in the frequency domain (those that require signal transformation), algorithms that operate in the time domain (those that process data straightforwardly in the time domain without needing any mathematical conversion), and algorithms that operate both in the time and frequency domains. Those that operate in the frequency domain include and are not limited to Fast Fourier Transform (FFT) while those that operate in the time domain include and are not limited to Kalman Filter (KF), Moving Average Filter (MAF), Recursive Least Square (RLS). Additionally, signal processing techniques that operate in both the time and frequency domains include and are not limited to Wavelet Transform (WT), Short Time Fourier Transform (STFT), etc. A detailed comparison of the different signal processing techniques with their advantages and disadvantages is found in [128].

2.6.3.1 Fast Fourier Transform

Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain. The decomposition of a sequence of values into components of different frequencies, which is performed by discrete Fourier transform (DFT), is useful in many fields. However, computing it directly from the definition is often too slow to be practical especially for large datasets. Fast Fourier Transform (FFT) algorithm rapidly computes the DFT of a sequence thus reducing the number of computations needed. FFT is one of the most utilized and oldest signal processing technique used in applications such as Structural Health Monitoring (SHM) because of its ease of implementation and efficiency of analysing stationary signals [128]. In the area of leak detection, FFT has been used in a myriad of studies for processing the leak signals detected by pressure transducers and/or vibro-acoustic sensors. Some studies that have used FFT for leak detection include: [54], [104], [111], [129].

2.6.3.2 Wavelet Transform

The Wavelet Transform (WT) is similar to FFT (or much more to STFT) with a completely different merit function. The main difference between the two is that FFT decomposes the signal into the frequency domain whereas the WT decomposes the signal into both the time and frequency domain. That is, whereas FFT provide good frequency resolution, but no time resolution, WT provides both good time and frequency resolutions of a signal [130]. This inherent feature of the WT has contributed to its use in a myriad of engineering applications especially in SHM, as it provides a time–frequency representation of the signal through time and scale window functions [128]. Just like FFT, WT has been applied in a lot of studies involved in leak detection and localisation in water pipelines. Some studies that have used WT for leak detection include: [40], [54], [131].

2.6.3.3 Time Series Analysis

Time series analysis is a statistical method that can analyse data sequences for the purposes of model identification, parameter estimation, model validation and prediction [132]. Time Series (TS) models develop an approximate mathematical model from input and output measurements recorded for a certain phenomenon [128] and can be used for prediction, signal extraction and

decision making for time series data (which is a record of a phenomenon irregularly varying with time). They can be classified into stationary and nonstationary, Gaussian and non-Gaussian or linear and nonlinear TS models. Linear TS models are among the most utilized techniques for signal processing and condition assessment of structures under dynamic loading owing to the fact that they can be implemented easily and are considered an efficient technique for modelling linear systems [128]. When considering stationary and linear TS, the most popular models are: Autoregressive (AR) model, Moving Average (MA) model, Autoregressive Moving Average (ARMA) model, as well as their variations such as Autoregressive Vector (ARV) model, Autoregressive model with eXogenous inputs (ARX), Autoregressive Moving Average with eXogenous input (ARMAX) , and Autoregressive-Autoregressive with eXogenous input (AR-ARX), and Autoregressive Integrated Moving Average (ARIMA) [128], [133]. Some representative studies that have used TS for leak detection include: [134], [135].

2.6.3.4 Kalman Filter

The Kalman Filter (KF) is a signal processing technique for filtering information known to be prone to error, uncertainty, or noise. KF is a time series model that can be used for prediction, signal extraction and decision making for time series data. It is considered as a powerful mathematical tool that can be used to predict the state of a dynamic system in the presence of uncertainty [136].

With respect to leak detection in WDN, KF and its variants, on the one hand have been applied as a state observer algorithm to determine parameters such as leak size and leak position. Torress et al. in [44] provides a detailed review of KF used as a state observer technique for leak detection in water pipelines. Other studies such as [101] and [26] use KF as a state observer. On the other hand, there exists a few studies in the literature dealing with leak detection in WWPM that have used KF as a signal processing technique for processing data gotten from vibro-acoustic sensors. Karray et al. in [38] used KF for filtering out noise from vibration measurements obtained from a FSR attached to the pipe while Kartakis et al. in [13] also used KF for removing noise present in the pipe surface vibration data collected by an accelerometer. In both studies, the KF is embedded within the sensor nodes, allowing the nodes to perform local processing. As such, both studies actually implement distributed computing within the WSN. However, both studies deploy a decentralized WSN architecture where the nodes forward their processed data to a cluster head and there is no

collaboration between neighbouring sensor nodes. In addition, none of the studies utilizes low-cost MEMS accelerometers for leak detection. More recently, Zhang et al. [50] also used KF alongside other algorithms such as Median and Interquartile Range Analysis and Cumulative Sum (CUSUM) methods in a real life WDN in South Australia that has 305 permanent accelerometers installed for monitoring the water pipeline. Though KF algorithm has been used for leak detection in WWPM, distributed KF algorithms have not been applied to WWPM, to the best of our knowledge.

In the following sub-section, we review some vibration-based WWPM studies available in the literature. The survey is based on the architectures of the sensor nodes, the type of sensors and pipe material used, and the location where the leak detection algorithms are implemented to analyse the leak signals.

2.6.4 A Review on Vibration-based WWPM Studies

Depending on where these computational algorithms (used for pre-processing of the leak signals, leak detection and leak localization) are executed, WWPM solutions can be classified as either centralized, decentralized or distributed. In order to have a good understanding of all the significant WWPM approaches available in the literature, we propose the criteria given in Table 2.2, which provides a description of the attributes we used to classify WWPM studies available in the literature. Our focus is on WWPM studies that monitor the pipe surface vibration as an indirect method of monitoring the pressure fluctuations caused by leaks in the pipeline and that make use of nonintrusive (vibration-based) sensors such as accelerometers, piezoelectric transducers, force sensitive resistor, etc. The algorithms used in these studies for processing the leak data (pipe surface vibration) fall in the data-driven, model-based, and signal processing methods of leak detection techniques shown in Figure 2.5.

Table 2. 2: List of criteria for comparing selected WWPM.

Criterion	Description
Parameter monitored	This is a feature of the pipeline system that is detected by the sensor and used for leak detection after processing.
Sensor	Type of sensor used in the study to detect leak signals
Pipe material	Type of pipe that was used in the study. It can be metallic (e.g., steel) or plastic (e.g., Polyvinyl Chloride (PVC))
Processing unit	This describes the on-board processing unit embedded in the sensor nodes
Pre-processing (PP)	Technique used for pre-processing (e.g., filtering) the leak signal
Leak Detection (LD)	Technique used for processing the leak signal to detect the presence or absence of a leak.
Leak Localization (LL)	Technique used for identifying the location of the leak.
Location of Processing	Processing can be done at the Base Station (BS), Fusion Centre (FC) or at the Sensor Node (SN).
Monitoring type	Classifies pipeline monitoring into Centralized, Decentralized, or Distributed based on the location where processing takes place. Centralized: all processing takes place at the BS. Decentralized: part of the processing (PP and/or LD) take place at the SN and/or FC. Distributed: all processing takes place at the SN.

We focus on 16 representative vibration-based WWPM studies and compare them based on the criteria listed in Table 2.2. Table 2.3 provides a summary comparison of selected studies in the literature that monitor pipelines using WSNs and make use of nonintrusive sensors.

The study of Stoianov et al. [42], referred to as PipeNet, is one of the pioneering vibration-based WWPM solutions that provides real-time leak detection. On the basis of an Intel commercial mote (Imote) composed of an ARM7 core, 64 kB of RAM, 512 kB of Flash, and a Bluetooth radio for communication, a laboratory pipe rig was built to demonstrate the detection and localization of leaks using acoustic and vibration data acquired from densely spaced hydrophones and accelerometers installed along the pipeline. Local processing was performed at each node by a Fast Fourier Transform (FFT) implementation combined with a compression, while cross-correlation was implemented at the central server as the leak detection and localization algorithm. The study provides a real-time solution for leak detection, but it is not energy efficient due to the high sampling rate and processing algorithms that were used.

In [41], Sadeghioon et al. present the design and development of a multimodal Underground Wireless Sensor Network (UWSN) for pipeline structural health monitoring. They developed a

sensor node consisting of a 16-bit microcontroller from Microchip, implementing their nano-watt XLP technology (PIC16LF1827), an eRA400TRS 433 MHz transceiver, two temperature sensors and one Force Sensitive Resistor (FSR) pressure sensor. According to the authors, the power consumption of the sensor nodes was minimized to 2.2 μW based on the collection of one measurement per 6 h in order to prolong the lifetime of the network. Two drawbacks could be highlighted from this work: the first one is its inability to perform real-time monitoring and the second one lies in the classic drawbacks from adopting a centralized approach for leak detection—reduced efficiency for a large-scale WSN, as it induces high latency and uneven energy distribution.

Martini et al. [39], in a series of tests in Bologna, used low-cost accelerometers attached to plastic pipes close to water meters in the city. They proposed to solve the problem of the high false alarm rate caused by the low accuracy of low-cost sensors and the inability of reliably detecting leakages in the midst of environmental noise, by taking measurements only during quiet times, for example, during the night when activities are reduced. However, one drawback with this approach is that does not operate in real time, as leaks cannot be immediately detected whenever they occur. Another reason is that it is difficult to find quiet times in certain areas such as city centres.

In [38], Karray et al. propose a solution called EARNPIPE which is comprised of a Leak Detection Predictive Kalman Filter (LPKF) and Time Difference of Arrival (TDOA) to detect and locate leaks. The study implemented a decentralized cluster-based WSN architecture for monitoring. The data collected from sensors were filtered, analysed and compressed locally with the same Kalman Filter (KF)-based algorithm and then forwarded to the cluster head to detect and localize the leak. A laboratory testbed was set with plumbing components and a network was deployed, consisting of nodes composed of Arduino Due board (with an ARM cortex M3 microcontroller inside), FSR sensors used for measuring pressure and Bluetooth for communication. In this work, the high consumption of the Arduino Due board combined with the power hungriness of Bluetooth communication leads to high consumption profiles and thus not appropriate for battery-powered WWPM systems.

Ismail et al., in [37], [49], [58], presented the development of a WPM using low-cost off-the-shelf components. The experimental setup consisted of low-cost vibration sensors such as MPU6050, ADXL335 and MMA7361 sensors for the measurement of vibration occurring along the pipes, an Arduino Uno and an XBEE module for wireless transmission to a centralized decision support system. Their work showed that low-cost MEMS accelerometers with multiple axis can be effective for leak detection in plastic pipes. Their solution was capable of distinguishing a leak from a non-leak for a leak coming from a 1-mm hole when the pressure was above 58.8 kPa. The drawbacks of this solution include the high rate of false alarms and the fact that it does not operate in real-time.

In [13], Kartakis et al. presented an end-to-end water leak detection and localization system, which exploits edge processing and enables the use of battery-powered sensor nodes. The proposed system combined a lightweight edge anomaly detection algorithm based on a Kalman filter and compression rates and a localization algorithm based on graph theory. It was validated by deploying nonintrusive sensors measuring vibrational data on a lab-based water test rig that had controlled leakage and burst scenarios implemented. The sensor nodes were based on Intel Edison development boards (embedding a dual-threaded Intel Atom CPU at 500 MHz and a 32-bit Intel Quark microcontroller at 100 MHz, 1 GB LPDDR3 POP memory as RAM and 4 GB eMMC as flash storage) and NEC Tokin ultra-high-sensitivity vibration sensors. The main drawback of this work is that the choice of commercial element (Intel Edison board) that constitutes the sensor node is not really a WSN node stricto sensu since it belongs to the Raspberry device class, with an energy efficiency and cost effectiveness that are beyond the specifications corresponding to WSN performances. In addition, the sensor used for monitoring the pipe surface vibration is not a low-cost sensor. However, even though our study is focused on using low-cost, low-accuracy vibration sensors for reliable leak detection, this article shows that edge computing has an emerging presence in the field.

Table 2. 3: Summary comparison of some selected studies in WWPM.

Ref	Monitored Parameter	Sensor	Processing Unit	Pipe Material	Pre-Processing Technique	Leak Detection Algorithm	Leak Localization Algorithm	Location of Processing	Monitoring Type
[13]	Pipe's surface vibration	Vibration Sensor (NEC Tokin)	Intel Edison development boards	N. A	Kalman filtering	Compression rates analysis	Graph-based technique	PP: SN LD: SN LL: BS	Decentralized
[37]	Pipe's surface vibration	Accelerometer (MPU6050, ADXL335 and MMA7361)	Arduino Uno (ATmega328P)	Plastic (polyethylene)	N. A	Offline analysis	N. A	PP: N.A LD: BS LL: N. A	Centralized
[38]	Pressure	Force sensitive resistor	Arduino Due (ARM cortex M3)	Plastic (polyethylene)	Kalman filtering and compression	Predictive Kalman Filter	Time of arrival difference	PP: SN LD: SN & FC LL: FC	Decentralized
[42]	Acoustic signals and pipe's surface vibration	Hydrophones and accelerometers (ADXL203EB)	Imote (ARM7)	Plastic (Polyvinyl Chloride)	Fast Fourier Transform (FFT) and compression	Acoustic leak detection technique	Cross-correlation	PP: SN LD: BS LL: BS	Decentralized
[41]	Pressure	Temperature sensor and Force sensitive resistor	Microcontroller (PIC16LF1827)	Plastic (Polyvinyl Chloride)	N. A	Relative pressure change	N. A	PP: N.A LD: BS LL: N. A	Centralized
[113]	Acoustic signals	Acoustic sensors	N.A.	Metallic	N. A	Acoustic emission technique	Cross-correlation method	PP: N.A LD: BS LL: BS	Centralized
[54]	Pipe's surface vibration	Accelerometer (KB12(VD))	N.A.	Plastic (polyethylene)	Moving average	Fast Fourier Transform, Wavelet Transform, Power Spectral Density and Cross Spectral Density	N. A	PP: BS LD: BS LL: N. A	Centralized

Ref	Monitored Parameter	Sensor	Processing Unit	Pipe Material	Pre-Processing Technique	Leak Detection Algorithm	Leak Localization Algorithm	Location of Processing	Monitoring Type
[127]	Pipe's surface vibration	Piezoelectric transducer	N.A.	Plastic (Polyvinyl Chloride)	Amplification	Amplitude thresholding and FFT	Localization based on leak index	PP: BS LD: BS LL: BS	Centralized
[39]	Pipe's surface vibration	IEPE accelerometer	N.A.	Plastic (polyethylene)	Signal filtering and amplification	Standard deviation computation	N. A	PP: SN LD: BS LL: N. A	Centralized
[137]	Pipe's surface vibration	vibration sensor	microprocessor	N. A	N. A	Power Spectral Density and Cross Spectral Density	Modified Maximum Likelihood prefilter	PP: N.A LD: BS LL: BS	Centralized
[138]	Acoustic emission	Hydrophone	Teensy 3.6 (ARM Cortex M4)	Plastic (Polyvinyl Chloride)	Amplification and Analog-to-Digital conversion	Linear Prediction	Cross Correlation	PP: SN LD: BS LL: BS	Centralized
[62]	Pipe's surface vibration	Accelerometer (CT1010)	CC2530 (8051 microcontroller)	Aluminium-plastic composite Pipe	N.A.	Principal Component Analysis and Support Vector Machine	N.A.	PP: BS LD: BS LL: BS	Centralized
[45]	Pipe's surface vibration	Accelerometer (ADXL345)	N.A.	Plastic	N.A.	Support Vector Machine, Fine K-Nearest Neighbours), and Medium Decision Tree	N.A.	PP: BS LD: BS LL: BS	Centralized
[46]	Pipe's surface vibration	Accelerometer (ADXL362 and SD1521)	Microcontroller (MSP430FR6989)	N.A.	Cross-Spectral Density Analysis	Leak Detection Index	N.A.	PP: FC LD: BS LL: BS	Decentralized

Ref	Monitored Parameter	Sensor	Processing Unit	Pipe Material	Pre-Processing Technique	Leak Detection Algorithm	Leak Localization Algorithm	Location of Processing	Monitoring Type
[50]	Pipe's surface vibration	Accelerometer (Von Roll)	N.A.	Metallic (Cast iron)	Minimum noise level from sliding windows	Median and interquartile range analysis and Kalman Filter	N.A.	PP: BS LD: BS LL: BS	Centralized
[40]	Pressure	Pressure transducer	Waspmote (AT mega 128)	Metallic (galvanized iron pipes)	Moving average filtering	Wavelet Transform	N.A.	PP: SN LD: BS LL: BS	Decentralized

As can be seen from Table 2.3, most of the existing WWPM studies are either centralized or decentralized since they require processing at the base station in order to detect and localize leaks. Additionally, from the literature, studies like [35], [37], [37], [39], [42], [43], [54], have demonstrated that low-cost MEMS accelerometers can be used for detecting leaks in plastic pipes. Thus, we will focus our study on WWPM systems that use low-cost MEMS accelerometers for leak detection in plastic pipes. Another reason for focusing on plastic pipes is that in most developing countries WDNs are constructed from plastic pipes because of their lightweight that facilitates their deployment over large areas, protection from corrosion, and resistance to bursts resulting from hydraulic variations [45].

Lastly, in order to achieve our dual goal of achieving improved accuracy and extending the lifespan of WWPM system, we will in the next sub-section review the power management techniques that have been used in WWPM to prolong the lifespan of WSNs used for monitoring WDNs.

2.6.5 A Survey of Power Management Techniques for Extending the Lifespan of WWPM Systems

In the literature of WWPM, very few studies have tackled the issue of power consumption and sought ways to reduce the energy consumption so as to prolong the monitoring lifetime. However, the issue of energy consumption in WWPM systems is of paramount importance since a WWPM system is required to go for long periods of time unattended as a result of the fact the nodes may be difficult to access, especially in the case of buried pipelines. In such cases, the Pipelines are required to be monitored throughout their life span, which can extend into years. In this sub-section, we survey some power management techniques that were discussed in section 2.3 and how they have been applied to WWPM.

In [41], the authors used duty cycling with scheduled wake-up to reduce the power consumption and thus prolong the WWPM lifetime. The Power consumption of the sensor nodes was minimized to 2.2 μW based on taking one measurement every six hours, thus given the sensor node a theoretical lifespan of 100 years when powered with two AA batteries. Although the sleep/wake-up method used in this study achieves great energy reduction, the WWPM solution does not provide real-time monitoring. In a more recent study, Liu et al. [62] proposed a leakage triggered

networking method (radio controlled wake-up) to reduce the wireless sensor network's energy consumption and prolong the lifetime of their proposed WWPM system. The authors proposed the use of three types of control frames (i.e., join frame, active frame, and wave frame) to trigger the network according to the leakage detection results. The coordinator and routing nodes were consistently in the working state and the terminal nodes within the routing node network were sequentially working and sleeping. The terminal nodes only wake-up from sleep when they receive an active frame from the routing nodes. In [107], the authors fused duty cycling and data driven-based schemes for maximizing the information gain about the leak, as well as minimizing the power consumption. The duty cycling-based schemes on the one hand work by putting the nodes into sleep mode when there is no activity. The authors exploited this idea to minimize the energy consumption. On the other hand, the data driven-based schemes focus on how energy can be conserved through efficient data reduction and data acquisition. In their proposed solution, the duty cycling part provides the sleep-wake-up schedule for the nodes to minimize the sensing, communication and processing-related energies while the data driven part implements adaptive sampling where nodes closer to the leak location operate at higher sampling rate whereas nodes farther away from the leak operate at a lower sampling rate. However, this study employed centralized data processing to detect and localise leaks. The same author in another study [139] combined duty cycling and data driven approaches like hierarchical sensing and compression. The scheme relies on implementing hierarchical sensing by using vibration sensors of different sensitivities to detect vibrations due to a leak [140], and on exploiting duty-cycling, and wavelet-based signal compression, in order to reduce sensing, computation and communication energies. However, the results were validated only via simulations.

Rashid et al. [40], used clustering and in-network processing to reduce the energy consumed in the network. The authors used wavelength transform and moving average filter to implement in-network processing within the WSN. By integrating the signal processing algorithm in the sensor nodes for distributed event detection and by performing aggregation on data within the cluster, the energy consumed in the network is far less than when all readings are sent to the base station in a centralized network. [46] and [38] also implemented clustering whereby sensor nodes acquire vibration data from the pipe and transmit to a closely located cluster head node which performs

local processing and finally transmits to a centralized base station that performs the leak detection decision.

A mobile sink approach is proposed in [141], to reduce the number of multi-hop communications involved in linear WSNs monitoring pipelines and to significantly reduce the energy consumption used in data transmission. The study presents a framework for monitoring linear infrastructures, where data collection and transmission in the linear WSN is done using Unmanned Aerial Vehicles (UAVs).

Elleuchi et al. in [61] proposed a novel heterogeneous two-tiered routing model that uses DEEC and RPL to achieve the routing task for WWPM. RPL was applied in the upper tier while DEEC is executed in the lower tier. The simulation results revealed the advantages of using the two routing protocols at the same time in decreasing energy consumption and extending the lifetime of a WWPM system. Their solution was however based on a centralized WSN architecture.

Kartakis et al. [13], presented an end-to-end water leak localization system, which exploits compression as the data reduction approach for conserving energy consumption. The proposed system combined a lightweight edge anomaly detection algorithm based on Kalman filtering on compression rate stream. The results revealed that the proposed solution reduced the communication by 99% compared to the traditional periodic communication, thus enabling the use of battery-powered sensor nodes for an extended period.

2.7 Summary and Identified Gaps in Knowledge

2.7.1 Summary

Achieving accurate real-time leak detection while preserving the lifetime of the WWPM system for a long period of time is a major challenge in WWPM. As has been shown in the literature, most of the WWPM systems available are centralized and decentralized. The centralized systems tend to consume much power and do not permit real-time monitoring thus making them not an optimal solution for leak detection in WWPM. Contrary, very few WWPM studies provide a fully distributed solution for leak detection. However, in recent years, there has been an increase in the number of studies advocating for the development distributed solutions [10], [34].

In addition, the issue of energy conservation is of paramount importance in WWPM. However, most WWPM studies focus on leak localization while neglecting the energy consumption of their proposed WWPM solutions. As such, there are very few studies in the literature that concern themselves with the evaluation of the energy consumption of their proposed WWPM solution and means on how to conserve energy in order to extend the lifespan of the WWPM system. Additionally, most of the studies that have treated energy consumption of their proposed WWPM solutions have been based on simulations and very few studies go to the extent of performing physical experiments on either a laboratory testbed or a real WDN.

Recently, studies seeking to achieve reliable leak detection in plastic pipelines using nonintrusive sensors like accelerometers have become popular, making WWPM using accelerometers a very active area of research. From the literature, most of the studies have proposed the use of high accuracy accelerometers and powerful signal processing algorithms for reliable leak detection. However, such solutions are not ideal for the low-cost and low-power requirements of WWPM systems as they neglect both cost and energy efficiency. Even though a number of recent studies have proposed the use of low-cost MEMS accelerometers for leak detection in plastic pipes. The combined issue of leak detection accuracy and power consumption have not been well treated. To the best of our knowledge, no study has investigated or applied redundant distributed data fusion algorithms such as DKF (which have been extensively used to increase the accuracy of tracking systems) in the area of WWPM so far.

Furthermore, being aware of the fact that the choice of components that constitute a sensor node affects the cost, power consumption and overall performance of the node, we noticed in the literature that few WWPM studies actually paid attention to the choice of components that constitute their sensor nodes. In order to achieve the objectives of low-cost, low-power, real-time and high reliability (which are conflicting requirements) in WWPM systems, it is required to properly scrutinise the selection of COTS elements used as building blocks of sensor nodes.

Thus, to provide a reliable, real-time, fully distributed and low-power consumption WWPM solution for monitoring plastic pipes, three things will be needed (hypothesis):

1. A sensor node with sufficient computing resources to perform in-situ processing while maintaining low-power consumption by efficiently implementing duty cycling.
2. Sensors to reduce the sensing energy via the implementation of hierarchical sensing.
3. A signal processing technique which is computationally less intensive and has both the capability of predicting sensed data so as to reduce communication cost and the ability to perform distributed data fusion so as to increase the accuracy of leak detection.

2.7.2 Identified Gaps in Knowledge

Following the literature search that has been conducted, the identified gaps in knowledge are summarised below:

1. While several studies in the literature are involved in providing leak detection in WWPM systems, most of the solutions either employed centralized or decentralized data processing to detect leaks. To the best of our knowledge, little work has been done to implement a real-time and fully distributed solution for leak detection in plastic pipes using low-cost MEMS accelerometer.
2. While most WWPM studies focus on leak detection and leak localization, very few studies have evaluated the power consumption of their solution and sought ways to reduce the energy consumption so as to prolong the monitoring lifetime.
3. No study has combined the implementation of duty cycling, hierarchical sensing, and data prediction energy conservation techniques at the sensor node level to prolong the lifetime of WWPM systems used for monitoring plastic WDNs.
4. No study has investigated a fully distributed WWPM solution based on DKF to improve the leak detection accuracy of plastic pipes monitored by low-cost MEMS accelerometers and evaluated its power consumption.
5. While several studies have evaluated the performance of different DKF algorithms in target tracking applications, no study has evaluated both the leak detection performance and energy consumption of DKF algorithms in the context of WWPM of plastic WDNs, by using both simulations and physical experiments on a laboratory testbed.

In order to fill in the gaps in knowledge identified above, we propose to answer the following research questions derived from our specific objectives in Section 1.4 of Chapter 1.

1. Which COTS components will be used to meet the requirements of a low-cost and low-power sensor node that has sufficient computing resources to perform in-situ processing?
2. How will the sensor node be designed, implemented, and configured so as to achieve high computational capacity and low power consumption?
3. Why is the KF a suitable signal processing technique for in situ processing of vibration data measured from plastic pipes using low-cost MEMS accelerometers?
4. How will distributed data fusion affect the leak detection accuracy?
5. How do the different distributed data fusion techniques affect the leak detection accuracy and the power consumption of the WWPM system?
6. Which DKF algorithm is optimal for real-time leak detection in plastic WDNs using WWPM systems composed of a network of low-cost MEMS accelerometers?
7. How will hierarchical sensing and duty cycling affect the sensor nodes energy consumption and the WSN lifetime?

Subsequent chapters of this thesis will provide answers to the research questions highlighted above.

Chapter 3

Sensor Node Design

The objective of this chapter is to determine the specifications of a cost-effective and low-power node with sufficient computing power (required for in-node processing) that will be constructed from low-cost commercial off-the-shelf (COTS) components as building blocks. This is important because we intend to achieve real-time leak detection by implementing all the processing required for leak detection within the sensor node and without the need for long distance communications via multiple hops to a base station. Thus, it is required that the processing unit of the sensor node have sufficient computing resources to perform the required computation and also consume less energy since the nodes are supposed to be battery-powered. Moreover, given that numerous sensor nodes will be required for monitoring plastic WDNs (since vibration signals do not go far in plastic pipes), it is necessary that the sensor node be low-cost.

In this chapter, we present both the hardware design and the configuration of a custom sensor node and a power measurement device which we will use for the evaluation of our proposed fully distributed WWPM solution on an experimental setup. Firstly, we find an answer to the question of which COTS components will be optimal for achieving a low-cost and low-power sensor node with sufficient computing capacity required for in-node processing. Next, we determine what will be the specifications of a sensor node suitable for providing a fully distributed, real-time, and low-power WWPM solution for monitoring plastic WDNs? To provide answers to the questions, we start the chapter by first performing a general overview of the constituent parts of a WWPM system and then review existing WSN hardware platforms (both commercial and research) with a focus on the processing and communication units. We later review low-cost MEMS accelerometers that can be used for measuring pipe surface acceleration on plastic pipes. Subsequently, we discuss the selection of the appropriate components for our custom sensor node and the reason for choosing them, based on our specific application requirements. This is then followed by a discussion on the design and configuration of the sensor node. The chapter ends by presenting the design and

configuration of a custom power measurement device for measuring the power consumption of sensor nodes on a laboratory testbed during the physical experiments.

3.1 An Overview of the Constituent Parts of a WWPM System

A WWPM system consists of sensor nodes placed along with the pipeline network at specific points. Like any WSN application, it has both a sensor layer and a cloud layer, as depicted in Figure 3.1.

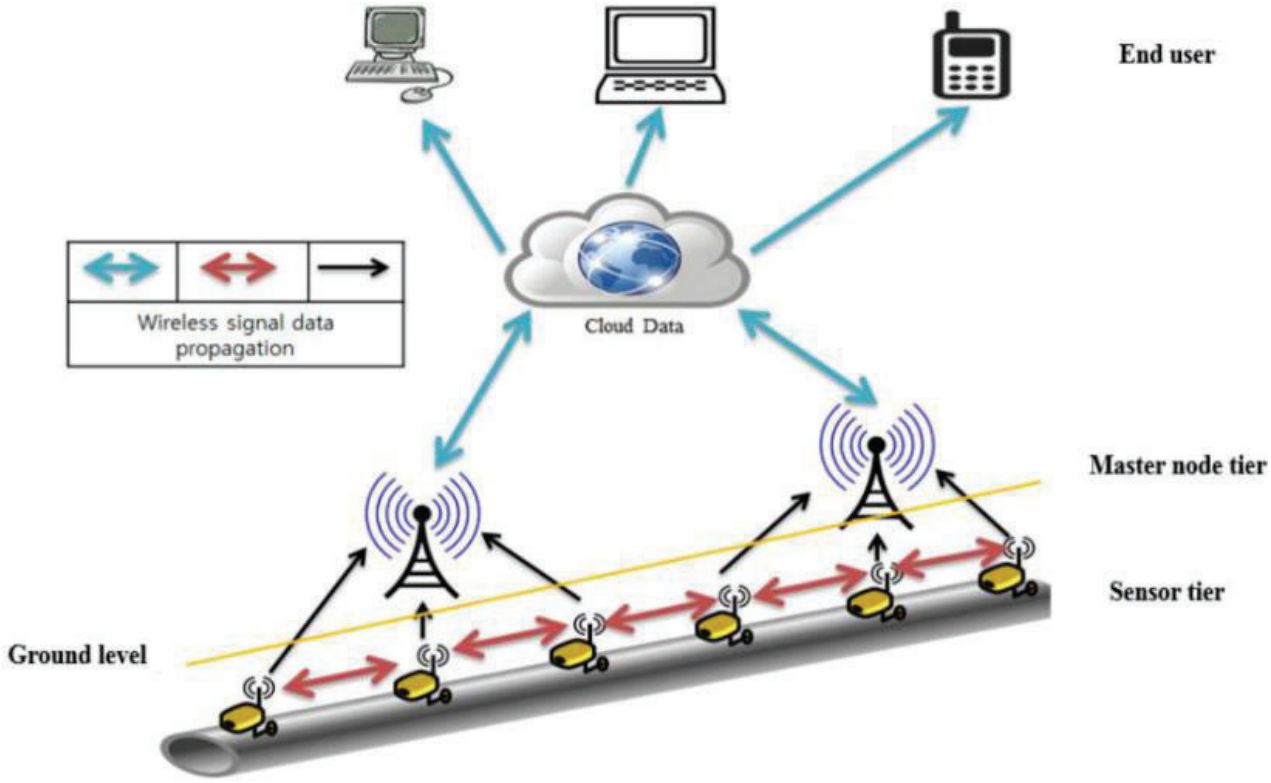

Figure 3. 1: Architecture of a WWPM system **[142]**

The cloud layer traditionally consists of algorithms that process the data collected from the sensor layer and databases that store the processed information for online visualisation [38]. The measurements collected by the sensor nodes get to the base station (sink) via either multi-hop communications or with the help of cluster heads in a centralized approach. The base station thus serves as an interface between the cloud layer and the sensor layer. The sensor layer, which serves

as an interface between the physical world and the digital world, consists of sensor nodes, with each node having the components depicted in Figure 3.2.

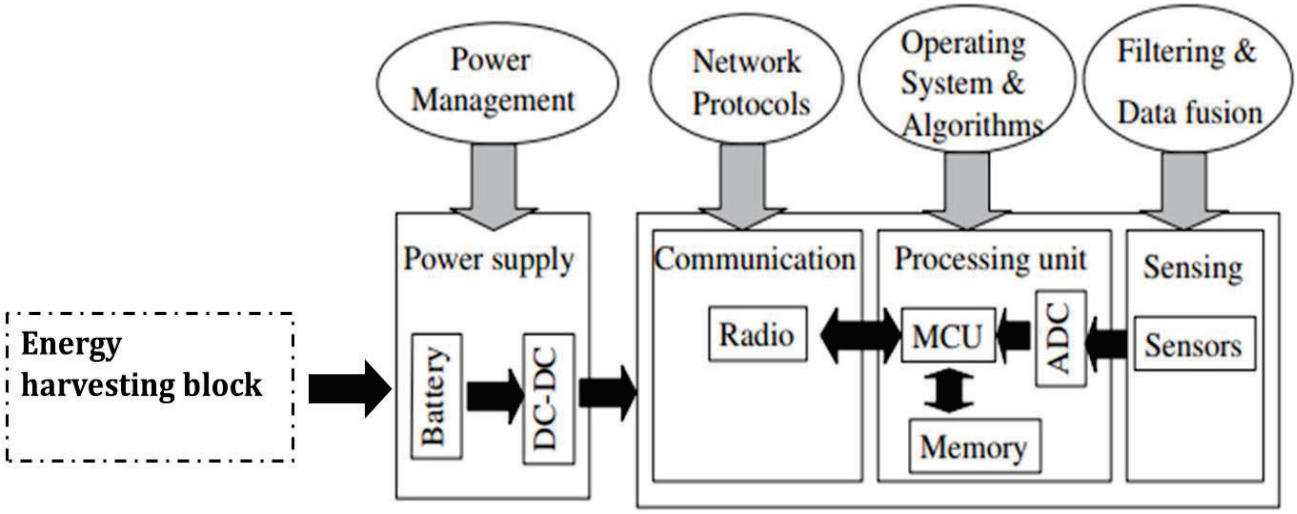

Figure 3. 2*:* Main hardware components of a wireless sensor node. Adapted from **[143]**

Given that this thesis focuses only on the sensor layer of the WWPM system, we will limit our discussion on the sensor nodes that constitute the sensor layer, since the cloud is not necessary. The components present at each sensor node include:

1. **Processing unit:** is the core of a wireless sensor node and it is involved with the collection of data from the sensors, processing this data (data filtering, data compression, data aggregation, data routing, etc.), deciding when and where to send it, reception of data from other sensor nodes, and setting of actuator's behaviours (if they are present). It has to execute various programs, ranging from time-critical signal processing to communication protocols of application programs [144]. Categories of processing and control units used in a sensor node include: Microcontroller (MCU), Digital Signal Processor (DSP), Field Programmable Gate Arrays (FPGA), and Application Specific Integrated Circuits (ASIC) [4].
2. **Communication unit or Transceiver:** is in charge of sending and receiving packets to or from other sensor nodes in the network via wireless communication. The transceiver has different modes of operation which include: transmitting, receiving, and idle/sleep modes, with each state consuming a different amount of energy [144]. The choice of the

communication unit is very crucial in determining the sensor node's energy consumption, since this unit consumes the highest energy compared to the processing and sensing units [1], [17].

3. **Memory unit:** A wireless sensor node should have sufficient memory space to perform the needed tasks. Typically, the on-chip memory of the MCU(RAM and EEPROM) usually serve as the memory unit [144]. In addition, some sensor nodes are equipped with flash memories, which are used to expand their storage capacity. The flash memories are used because of their ability to provide high capacity at a considerably low-cost [4], and can be used to store application related data or programs.
4. **Power supply unit:** The operations (sensing, data processing, and communication) of a sensor node consume power. Sensor nodes are usually powered via energy stored in batteries or capacitors in applications that require deployment in areas without access to the power grid. The batteries used can either be rechargeable or non-rechargeable and can also be classified into nickel–cadmium (NiCd), nickel–zinc (NiZn), nickel metal hydride (Nimh) and lithium ion, depending on the electrochemical material used by the electrodes [145]. For long-lasting WSN applications where there is a need for sensor nodes to go for long periods unattended and without replacing their energy source, the limited energy storage capacity of batteries is not attractive. Currently, most sensor nodes are designed to have the optional ability to recharge their battery from energy harvested through scavenging techniques such as photovoltaics, temperature gradients, vibrations, pressure variations, the flow of air/liquid etc. [71], [144], [145]. In addition, sensor nodes also employ power management techniques such as duty cycling and dynamic voltage scaling to prolong the lifespan of the batteries.
5. **Sensing unit:** serves as an interface between the real world and the digital world. It is usually made up of transducer and signal conditioning (ADC, filter, and amplifier) parts. The transducer generates an electric signal proportional to the event or condition being monitored or measured [4] and the generated electric signal is typically converted to digital form using the ADC, since the processing unit of sensor nodes can only process digital data [145].

Now that we have discussed the constituent parts of a wireless sensor node, we review in the next section the different sensor node hardware platforms that exist in the literature to guide our selection of a sensor node to be used for our fully distributed WWPM solution.

3.2 A Review of WSN Hardware Platforms

3.2.1 Introduction

The architecture and technology of a sensor node are crucial in determining its cost, performance, and power consumption. The fact that a fully distributed solution for WWPM will require distributed computations, where the onboard processing capabilities of each sensor node are involved in processing the filtering, leak detection, and localization algorithms, necessitates that the sensor nodes operate more than just data collectors as were intentionally designed for, to full-fledged information processors. Besides, the requisite of low power consumption in a WWPM system to prolong its operational life will also be hugely affected by the components that constitute the sensor nodes. Thus, the choice of using either an MCU, DSP, FPGA or ASIC as the sensor node's processing unit depends on a trade-off between flexibility and performance. Figure 3.3 depicts the trade-off between flexibility and performance for the different processing platforms, with ASICs (which are entirely hardware-based architectures) providing the highest performance and MCUs (which are software-based architectures) providing the most flexible solution [67].

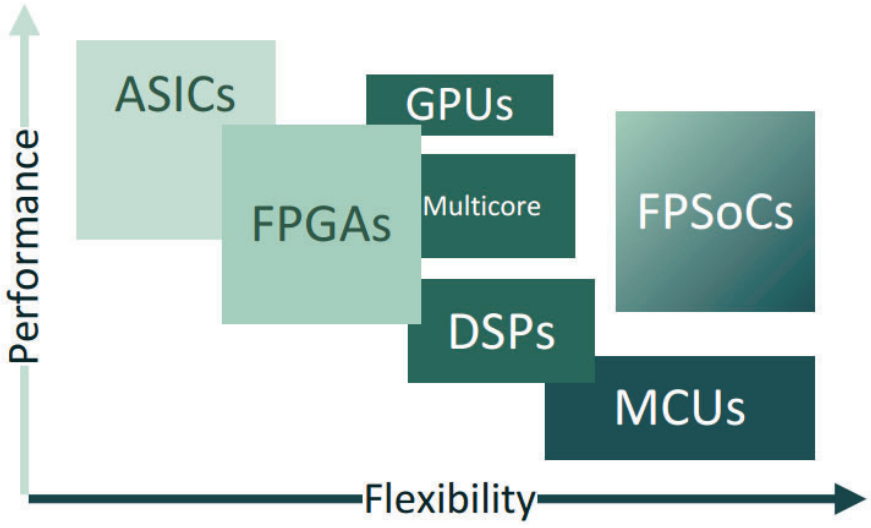

Figure 3. 3: Flexibility-Performance trade-off for different sensor node processing units **[67]**

1. **MCUs:** The increased use of MCUs in sensor nodes is because of their flexibility in connecting with other devices (like sensors), their instruction set being amenable to time-critical signal processing, their inherent capability of integrating ADC and DAC for conversion and timers used for control, and their low power consumption; they are also convenient in that they often have built-in memory [144]. Moreover, they can easily be programmed and can harness system software such as embedded operating systems in programming complex applications. Furthermore, their ability to operate in different power modes, which permits only certain parts of the MCU to be active at any given time, can be harnessed in the power optimization of the sensor node's power consumption [144]. Thus, MCUs are increasingly used in sensor nodes because of their reduced cost, low-power and easy interface [4], [67].
2. **DSPs:** While the utilization of DSPs as the processing units in the sensor nodes of some WSN applications (e.g., SHM) is possible, [144], [145], they are more optimized than general-purpose processors for applications requiring dedicated signal processing functions like digital filtering and feature extraction [4]. However, unlike MCUs they are more expensive and complex for WSNs environments with relatively simple and low signal processing requirements. Their power consumption is also high, making them not a suitable candidate for battery-powered sensor nodes. They are usually serve as coprocessors in sensor nodes rather than the main processor [4].
3. **FPGAs**: which are reconfigurable platforms, offer greater flexibility, speed, performance, and reliability and can be used in sensor nodes as a standalone processor, a coprocessor, a hardware accelerator, or as a reconfigurable unit [4]. The reconfiguration attribute of FPGAs permits them to be reprogrammed in the field after deployment to adapt to a changing set of requirements. However, compared to MCUs, the process of reconfiguration can take a long time and requires more energy.
4. **ASICs:** implement the required functionality using only hardware and provide better energy efficiency and performance. However, ASICs lack flexibility, and their functionalities can be offered in a less expensive and more flexible way via MCUs using software means [144].

There are several studies in the literature that review the different processing platforms used in sensor nodes. Notably, Karray et al. in [4] presented a comprehensive review of the current state-of-the-art background needed for the design of evolutionary nodes with a thorough analysis of the node design components and application trends. The paper classified existing node platforms based on the technology used, from simple MCUs to sophisticated high-processing platforms, providing a selective state-of-the-art review on sensor node platforms, architecture, and technologies. The authors also provided a detailed comparison of all the different architectures. According to the authors, selecting the most optimal node architecture is a very important step in determining the performance and lifespan of the node and the network as a whole.

In summary, ASICs are preferred for WSN applications that do not change over time, that require better energy efficiency and high-speed performance, and there is sufficient finance for investment. Conversely, MCUs are suitable for WSN applications where there is a need for flexibility and implementation of a low-cost solution. FPGA-based platforms can provide the best of the two worlds as they offer high processing capabilities, resulting in higher performance when compared to MCUs, and the ability to be reconfigured in run-time, providing better flexibility when compared with ASICs[67]. However, they are not easy to use and are more expensive than MCUs.

Now, given that one of the objectives of this thesis is to propose a low-cost fully-distributed WWPM solution, we will use an MCU as the processing unit of our sensor node since MCUs are software-based devices and they represent the most flexible platform, making them a widely used in simple embedded systems with low-budget requirements [4], [67], [144]. Thus, we will limit this review only to MCU-based WSN hardware platforms.

Concerning the communication unit of sensor nodes, Radio Frequency (RF)-based communication is by far the most relevant communication means in WSN as it best fits the requirements of most WSN applications such as providing relatively long-range and high data rates, acceptable error rates at reasonable energy expenditure, and does not require line of sight between the transmitter and the receiver [4], [144]. So, we will focus our survey only on RF-based transceiver modules.

In the following sub-sections, we will review the MCU-based processing units and RF-based transceivers of commercial and research wireless sensor nodes. This is to facilitate the selection of a low-cost and low-power MCU and RF transceiver to constitute the building blocks of the sensor node to be deployed on our WWPM system.

3.2.2 An Overview of Commercial and Research Sensor Node Platforms

The deployment of a WSN application first requires the sensor nodes to be developed. There exist a myriad of research and commercial sensor nodes available in the literature from which one can choose from. However, the choice of a specific sensor node is usually influenced by specifications such as application requirements, cost, size, power consumption, sensor interface, computing power, memory resources, etc., as the nodes vary in their underlying properties and capabilities. Examples of research and commercial hardware platforms used in WSN and IoT applications include: Mica, Mica2, MicaZ, TelosB, Imote2, Sun SPOT, ESP8266, ESP32, Raspberry Pi, etc.

Over the years, there have been several studies that have sought to classify hardware platforms used in WSN. In one of the studies, Hahm et al. [88] classified IoT devices into two categories namely: low-end and high-end devices, based on the performance and capability of the hardware. In a more recent study, Ojo et al. [89] classified IoT platforms into three categories by further splitting low-end devices into low-end and medium-end devices. The low-end, medium-end, and high-end are likened to embedded sensor modules, system on chip (SoC), and adapted general-purpose computers, which was the classification of WSN hardware platforms provided in earlier studies such as [146], [147]. Each category shows a different hardware setup which can be matched to diverse monitoring applications and entails a different set of trade-offs in the design choices.

- **Low-end devices:** These are devices that are constrained in terms of computing and memory of resources. Their memory footprint is so constrained that they are unable to run traditional operating systems [88], [89]. These platforms are assembled from COTS MCUs, with RAM and flash memories that are of the order of tens or hundreds of kilobytes and processing units incorporating an 8-bit or 16-bit architecture. However, recent years have seen an influx of devices supporting 32-bit architecture, and 64-bit architecture devices are expected in the future. In 2014, the Internet Engineering Task Force (IETF) [87]

standardized the classification of these low-end IoT devices into Class 0 (<< 10 kB of RAM and <<100 kB Flash), Class 1 (~10 kB of RAM and ~100 kB Flash), and Class 2 (~50 kB of RAM and ~250 kB Flash). In addition, these devices usually use IEEE 802.15.4-based transceivers for communication since they are required to be low-power and their power sub-system usually consists of a battery. Examples of low-end devices include Mica, Mica2, MicaZ, TelosB, Imote2, OpenMote-B, etc.

- **Middle-end devices:** They lie between low-end and high-end devices with respect to the constrain of memory and computing resources. These devices are usually developed as SoC, incorporating MEMS sensors, MCUs, and one or more wireless transceiver modules on a single chip. In addition, they usually have their clock speed and RAM in the order of hundreds of MHz and KB, respectively unlike low-end devices that have their clock speed and RAM in tens of MHz and KB, respectively [89]. Moreover, these devices usually have transceiver modules that implement wireless technologies such as Wi-Fi, Bluetooth, BLE, Thread, etc. In terms of power supply, these devices can be power supplied by a battery or by the mains power supply. However, since these devices are power-hungry, powering them using a battery is not a good option for cases where the nodes are required to be operational for long periods unattended. This is because the battery will be depleted very fast and will require constant replacement or recharging over short periods. Examples of middle-end devices include ESP32, Particle Electron, Argon, Neon, etc.
- **High-end devices:** These platforms are low-power PCs, embedded PCs, and some personal digital assistants (PDAs), including single-board computers such as the Raspberry Pi and smartphones [88], [89]. They have sufficient memory and computing resources required to run a traditional OS such as Linux, Windows 10 IoT Core or other operating systems developed for mobile devices. Their boards are also incorporated with standard wireless communication devices like Wireless LAN (IEEE 802.11) and/or Bluetooth (IEEE 802.15.1) and they support networking protocols like IP. Their high computing and high bandwidth communication ability enables them to be used as IoT gateways to implement edge computing. They consume a considerable large amount of energy and are conveniently powered by the mains power supply or other means that provide large capacity power supplies.

Table 3. 1: Comparison of Research and Commercial Sensor Node Platforms

Platform	Processing Unit	Communication Unit	Research group/Vendor	Cost ($)	Category
IRIS (2007)	Atmel ATmega1281	Atmel AT86RF23 radio	Crossbow	115.00	Low-end
Mica2 (2002)	Atmel Atmega128L	CC1000	Berkley/Crossbow	99.00	Low-end
MicaZ (2002)	Atmel Atmega128L	CC2420	Berkley/Crossbow	99.00	Low-end
TelosB/TmoteSky (2004)	TI MSP430	CC2420	Berkley Moteiv	104.00	Low-end
Waspmote (2016)	Atmel ATmega1281	ZigBee 3/ 802.15.4/ Wi-Fi	Libelium	210	Low-end
Imote2 (2008)	Intel PXA271 XScale	CC2420	Crossbow	300	Low-end
LOTUS (2011)	ARM Cortex M3	RF231	MEMSIC	300	Low-end
OpenMote B (2019)	ARM Cortex-M3	CC2538 + Atmel AT86RF215	Industrial Shields	166.84	Low-end
Arduino UNO WiFi Rev.2 (2018)	ATmega4809	Wi-Fi	Arduino	38.90	Low-end
ESP8266 (2014)	Tensilica L106	Wi-Fi	Expressif	18.95	Middle-end
Sun SPOT (2007)	ARM9	CC2420	Sun microsystems	750.00	Middle-end
ESP32 (2016)	Dual-Core Xtensa LX6	Wi-Fi + BLE	Expressif	19.95	Middle-end
CC2538 (2012)	ARM Cortex-M3	IEEE 802.15.4 (Thread, Zigbee)	Texas Instruments	14.68	Middle-end
nRF52840 (2019)	ARM Cortex-M4F	Wi-Fi + Bluetooth + BLE + IEEE 802.15.4/Zigbee/Thread	Nordic	24.95	Middle-end
NCS36510 (2018)	ARM Cortex-M3	IEEE 802.15.4	Onsemi	201.15	Middle-end
Particle Argon (2018)	ARM Cortex-M4F	Wi-Fi + Bluetooth + BLE + IEEE 802.15.4/Zigbee	Particle	27.92	Middle-end
Raspberry Pi 4 Model B (2020)	Quad core 64-bit ARM-Cortex A72	Wi-Fi + BLE + Gigabit Ethernet	Raspberry Pi	55.50	High-end
Samsung ARTIK 710 (2016)	Dual core ARM Cortex-A7	Wi-Fi + Bluetooth + BLE + IEEE 802.15.4/Zigbee	Samsung Semiconductor, Inc	44.13	High-end
BeagleBone Black Wireless (2014)	ARM Cortex-A8	Wi-Fi + Bluetooth	Texas Instruments	78.75	High-end

Table 3.1 provides a comparison of popular research and commercial sensor node hardware platforms, categorised as either low-end, middle-end, or high-end devices. The list is by no means exhaustive, but it makes mention of most of the popular sensor node platforms available in the literature and which have been used in both research and commercial applications. This is coherent with past studies such as [4], [89], [148] but with the addition of recent platforms that have emerged. It should also be noted that low-end devices are greatly used in WSN applications (where

nodes do not directly connect to the Internet) whereas the usage of middle-end and high-end devices is dominant in IoT applications (where the nodes directly connect to the Internet and make use of IP or IP-like protocols).

3.2.3 A Survey on MCU-based Processing Units for WSN Hardware Platforms

WSN hardware platforms based on MCUs have evolved from using 8-bit MCUs to 32-bit MCUs. The first generation of sensor nodes made use of 8-bit microcontrollers and examples include MicaZ, Mica2, IRIS, Imote, etc. [7]. They are not suitable for deployment in our fully distributed solution because of their limited computational performance (in terms of computing speed and the size of RAM and flash memory), which permits minimal or no processing of the collected raw data onboard. Subsequently, the combination of 16-bit MCUs such as MSP430 and the CC2420 radio transceiver led to the development of second-generation sensor nodes [7] with examples such as TelosB. These devices permit some level of storage and pre-processing at the sensor node level but are still not sufficient for performing the considerable amount of local processing required at the sensor node of a fully-distributed WWPM solution. Recently, third-generation sensor nodes, initially introduced by a generation of 32-bit MCUs based on ARM Cortex –M0/M0+/M3/M4 and PIC32MX [6] (e.g., Sun SPOT, Imote2, and LOTUS), have become popular as they provide high processing power and consume less energy. The first release of 32-bit MCUs was later reinforced by the emergence of the second release of low-power 32-bit MCUs (led by ARM Cortex M7, dual-core ESP32, and faster PIC32MZ) in the period between 2015 and 2016, which triggered the use of third-generation sensor nodes in most WSN applications requiring in-node processing [6]. Common features of this generation of MCUs include low power consumption, the integration of powerful digital signal processing units, support of both Wi-Fi and Bluetooth network connection and larger RAM and Flash memories necessary for performing complex processing on the collected data onboard [6].

Table 3.2 provides a comparison of the specifications of popular MCU-based processing units used in the sensor node platforms mentioned in Table 3.1.

Table 3. 2: Comparison of popular MCUs used in WSN hardware platforms

MCU (Microprocessor)	Bus Size (bits)	CPU Speed (MHz)	RAM	EEPROM	Flash	Power Consumption		
						Active (mA)	Sleep (µA)	Voltage (V)
Atmel ATmega1281	8	8	4 KB	4 KB	128 KB	10 - 14	1 – 7.5	3 - 5
Atmel Atmega128L	8	8	4 KB	4 KB	128 KB	17 - 19	< 25	3 - 5
MSP430G2553	16	16	512 KB	N.A.	16 KB	4 - 5	0.8 - 56	1.8 - 3.6
Intel PXA271 XScale	32	13–416	256 (SRAM) 32 MB (SDRAM)	N.A.	32 MB	31	390	3.2 – 4.5
Atmel SAM3X8E (ARM Cortex M3)	32	84	86	16	512 KB	100	2.5	1.6 – 3.6
STM32F102Cx (ARM Cortex M3)	32	48	10/16 KB	N.A.	64/128	8.6 -36.1	200	< 4
ATmega4809	8	20	6 KB	256 Bytes	48	1.2 – 11.4	0.6 - 16	1.8 – 5.5
ESP32	32	240	520 KB	448 KB	2 MB	20 - 68	10 - 150	2.2 – 3.6
STM32F415RG (ARM Cortex-M4)	32	168	192 KB	N.A.	1024 KB	2 - 87	3 - 4	1.8 – 3.6
Raspberry Pi 4B (ARM-Cortex A72)	64	1500	1-4 GB	N.A.	N.A.	600	N.A.	5.1

3.2.4 A Survey on RF Transceivers for WSN Hardware Platforms

The RF transceiver unit (responsible for exchanging data between individual sensor nodes) is the most power-hungry component of the wireless sensor nodes, making its selection crucial, especially in battery-powered wireless sensor nodes.

Table 3. 3: Comparison of popular RF transceiver units used in WSN hardware platforms

RF transceiver	Power consumption				Outdoor coverage range (m)	Operational frequency band	Max. Data Rate
	TX (mA)	RX (mA)	Idle (μA)	Sleep (μA)			
RFM TR1000	12	8.0	N.A.	0.7	< 91.44	916.5 MHz	1 Mbps
RFM TR1001	12	3.8	N.A.	0.7	< 91.44	868 MHz	115.2 kbps
Chipcon CC1000	26.7	9.6	74	0.2	< 300	315 / 433 / 868 and 915 MHz	76.8 kbps
Chipcon CC2420	17.4	18.8	426	20	< 100	2.4 GHz	250 kbps
Atmel AT86RF215	62	28	6.28	0.03	< 420	315 / 433 / 868/915 MHz and 2.4 GHz	50 kbps
Atmel AT86RF230	16.5	15.5	N.A.	0.02	< 500	2.4 GHz	250 kbps
Atmel AT86RF231	14	12.3	0.4	0.02	N.A.	2.4 GHz	2 Mbps
Nordic nRF903	29.5	22.5	600	1	< 1300	433 / 868/915 MHz	78.6 kbps
Nordic nRF2401	13	19	12	1	< 500	2.4 GHz	1 Mbps
Nordic nRF24L01+	11.3	13.5	26	0.9	< 500	2.4 GHz	2 Mbps
Digi XBee S2C module	33	28	N.A.	1.5	< 1200	2.4 GHz	250 kbps

Table 3.3 displays the properties of popular radio transceiver units used in the sensor node hardware platforms presented in Table 3.1. We review only radio chips that conform to the IEEE 802.15.4 standard since they are the most widely used in sensor node hardware platforms due to their low power and long-range transmission compared to Bluetooth (BT). BT (IEEE 802.15.1) and Wi-Fi (IEEE 802.11) radios are also used in sensor nodes, with the argument being that they allow easy interoperability with a range of existing devices such as mobile phones and laptop computers without the need for additional hardware. However, they provide this interoperability at the cost of high energy consumption, making them not suitable as transceivers for battery-powered WSN applications.

The IEEE 802.15.4-based RF transceivers have varying characteristics such as operational frequency band, outdoor coverage range, data rate, cost, and current consumption at different operational (transmitting, receiving, idle, sleep) states. Our selection of an IEEE 802.15.4-based radio chip as the communication unit of our sensor node was influenced by attributes such as current consumption, price, and data rate.

3.2.5 Selected RF Transceiver and MCU for the Custom Sensor Node

After reviewing research and commercial sensor node platforms in sub-section 3.2.2, we did not find a specific sensor node platform that met all our requirements (low-cost, low-power and high computing capacity). But after reviewing several MCUs in sub-section 3.2.3 and RF transceivers in sub-section 3.2.4, we found out that we could construct a custom sensor node that conveniently meets all our requirements by adapting an MCU and RF transceiver selected from the reviews conducted in sub-sections 3.2.3 and 3.2.4. The advantage of building a custom node from COTS is that it leads to the development of a cheaper node that arguably provides comparable or improved performance in terms of power consumption, communication, and processing resources when compared to research and commercial nodes designed specifically for WSN applications [148].

From Table 3.3, it is evident that the nRF24L01+ transceiver compared to other IEEE 802.15.4-based transceivers operating at the 2.4 GHz band, exhibits current peaks in RX/TX modes lower than 14 mA (one of the lowest consumptions on the market), a sub-μA power-down mode, advanced power management, and a supply voltage extending from 1.9 to 3.6 V. The nRF24L01+ is hence a true ultra-low power solution enabling months to years of battery life from coin cell or AA/AAA batteries. Besides, compared to other IEEE 802.15.4-based radio chips, the nRF24L01+ transceiver is low-cost. Finally, the enhanced ShockBurst feature of the nRF24L01+ enables it to achieve high data rates (up to 2 Mbps) at lower power consumption. The burst mode is particularly interesting for distributed computing, which involves only short-distance communications between neighbouring nodes. Consequently, all these features make nRF24L01+ radio the ideal IEEE 802.15.4 compliant radio transceiver unit for use in our custom sensor node which is required to be low cost and consumes less energy.

From the Table 3.2, it can be seen that the ESP32, which was released in the last quarter of 2016 is one example of a powerful, low-cost, and low-power MCU. It incorporates a double-core 32-bit Xtensa LX6 microprocessor and an ultra-low-power (ULP) coprocessor. The ULP coprocessor consumes minimal current (between 10 μA~150 μA) when the core is sleeping and can be used for simple control. This ULP feature makes the ESP32 a suitable processing unit for a sensor node that will be battery-powered. Thus, a great advantage of the ESP32 is its ULP coprocessor which

gives it the ability to achieve ultra-low power consumption for most WSN operations and only uses the dual-core processor when there is a need to perform powerful computations. Moreover, the ESP32 has a feature that permits the remaining charge of its battery supply to be estimated by reading the battery voltage from one of its analogue pins. This feature makes the ESP32 suitable for energy-aware applications as it can compute its energy consumption online and thus regulate its operations. Another advantage of the ESP32 is its lower cost, which is an attractive feature for sensor nodes that are to be deployed in a developing country's WWPM system. Besides, the ESP32 also incorporates Wi-Fi and Bluetooth modules which makes it IoT compatible. Incorporating such an MCU into the sensor node's hardware will cause the nodes to evolve from simple sensors to powerful computing platforms [4].

Given that our fully distributed solution entails the embedding of data processing within the sensor nodes, an MCU with higher computational power will be ideal. Based on the classification of sensor nodes by Ojo et al. [89], the ESP32 SoC is a middle-end IoT device, and thus not suitable for battery-powered WSN applications. Even though SoC solutions have significantly reduced the amount of power consumption in the sleep mode, not much has been achieved in the transmission and receiving modes. This explained why we did not use the BLE and Wi-Fi transceiver modules of the ESP32 for communication between sensor nodes, but we instead had to perform a survey on low-power IEEE 802.15.4 compliant radio chips, and selected the low-power nRF24L01+ transceiver module. The reason for the low power consumption in the sleep mode is because the trend in this new generation of sensor nodes is to use low power processors that provide very low energy consumption when the node is inactive and to use the high-power processors when the node is active. Accordingly, the ESP32 incorporates a 32-bit dual-core Xtensa LX6 processor with a processing speed of up to 240 MHz as the core processor (operational when the ESP32 is in the active state) and a ULP coprocessor (operational when the ESP32 is in the inactive state). The core processor can achieve local processing while the ULP coprocessor can achieve real-time monitoring while maintaining low-power consumption by putting the core processor in an inactive state. Thus, by putting the core processor of the ESP32 in the inactive state regularly while making use of the ULP coprocessor, shutting down the Wi-Fi and BLE transceiver modules of the ESP32, and using a low-power transceiver module like the nRF24L01+ for communication, we can have

a custom sensor node with high computing power and whose energy consumption is similar to a low-end node. This custom node composed of the ESP32 and nRF24L01+ is low-cost and has sufficient computing power required for implementing low-power distributed computing in WWPM. Additionally, the ability of this custom node to be aware of its energy consumption makes it suitable for our computing energy-aware custom node. This is because it can adjust its operations to adapt to its currently available energy. Finally, the ESP32 being a middle-end device has been used extensively in IoT applications and not in WSN applications. However, using the ESP32 as the processing unit for our custom sensor node can also establish a scenario where we can examine the possibility of using IoT devices in WSN and their effect.

Now that we have selected an MCU and RF transceiver to serve as the processing unit and communication unit, respectively, for our custom sensor node, it is time for us to select a low-cost MEMS accelerometer to serve as the sensing unit of our custom node. In the next section, we will review low-cost MEMS accelerometers.

3.3 A Review of Low-cost MEMS Accelerometers

Vibration-based methods for monitoring the health of water pipelines are popular in the literature of WPM because of their ease of installation/maintenance, low cost, and low-power consumption. Specifically, several WPM studies have used accelerometers for leak detection. To measure vibrations of small magnitude such as pipe surface vibrations will require an accelerometer of high sensitivity. Also, the noise floor level of the accelerometer is required to be very low since a high noise floor level would mask the low amplitude vibration signals, thereby preventing their detection [46]. Examples of accelerometers with high sensitivity, low noise floor level, and wide frequency response range are the Brüel & Kjær (B&K) accelerometers [149]. However, they are very costly, making them not a very good candidate for a low-cost WWPM solution. MEMS accelerometers are attractive for our proposed WWPM solution because of their low prices, low power consumption, small-sizes, and suitability for measurement of amplitudes of applications of low frequencies (which is the case of WPM) [150]–[152].

The study by Stoianov et al. [42] is one of the pioneer studies that made use of MEMS accelerometers for leak detection in WWPM systems, though this study was carried out on a

metallic pipeline. Given that, the WDN of most developing countries is made up of plastic pipes and despite the challenges of detecting leaks in plastic pipes by using accelerometers that we presented in sub-section 2.6.2 of Chapter 2, studies such as [39], [54] have demonstrated the use of accelerometers in detecting leaks in plastic pipelines. However, these studies made use of high-cost and high accuracy piezoelectric accelerometers which makes them not economically optimal for low-cost WWPM solutions. In the past few years, the use of low-cost MEMS accelerometers for leak detection has been demonstrated in studies such as [35], [37], [46], [58]. These studies made use of both analogue and digital accelerometers for leak detection. In a recent study, Tariq et al. [43] presented a comprehensive review of MEMs-based leak detection studies available in the literature.

MEMS accelerometers can be categorised as either analogue or digital depending on the format of the output they deliver. WWPM solutions available in the literature have used both types of MEMS accelerometers for leak detection. One of the differences between the two is that the performance of digital MEMS accelerometers is not affected by other circuit components, unlike analogue MEMS accelerometers whose performance is dependent on the features such as the resolution of the external ADC [151]. The use of an external ADC with a higher resolution will, on the one hand, improve the sensitivity of the analogue accelerometer while, on the other hand, it will also increase the cost of the sensor node. Even though the ADC of the microcontroller (serving as the processing unit of sensor node) can be used, its low resolution (12 bits for the ESP32) will degrade the sensitivity of the analogue accelerometers when compared to the digital ones, which have dedicated internal ADC within their breakout boards. For this reason, we focus this survey primarily on digital MEMS accelerometers.

Because the performance of the accelerometer used for measuring the pipe surface acceleration can affect the quality of the measurements and the overall accuracy of the leak detection system, it is imperative to pay attention to the selection of the accelerometer. This survey focuses on low-cost commercially available MEMS accelerometers that are readily available in the market.

Table 3.4 presents a survey of digital COTS MEMS accelerometers that can be used for measuring the pipe surface acceleration of plastic pipes. The characteristics used for comparing the sensors

include bandwidth, sensitivity, resolution, current consumption, sensing range, and cost. For boards like the MPU6050, LSM9DS1, and BMI160 (which are Inertia Measurement Units (IMUs), consisting of both an accelerometer and gyroscope on the same IC), we focus only on the accelerometer. The data presented in Table 3.4 for such boards is based on the accelerometer only, without considering the gyroscope.

Table 3. 4: Survey of popular digital COTS MEMS Accelerometers

Accelerometer	Resolution	Bandwidth (Hz)	Sensitivity (LSB/g)	Sensing Range	Noise floor level (µg/√Hz)	Current Consumption (µA)	Voltage Supply (V)	Cost ($)
ADXL344	13	0 - 1600	4096	±2g, ±4g, ±8g, ±16g	530	23	1.7 – 2.75	38.54
ADXL345	13	0 - 1600	4096	±2g, ±4g, ±8g, ±16g	430	23	2.0 – 3.6	36.88
ADXL362	12	0 - 200	2048	±2g, ±4g, ±8g	550	1.8	1.6 – 3.5	40.10
BMI160	16	5.06 - 684	32,768	±2g, ±4g, ±8g, ±16g	180	180	1.71 – 3.6	25.48
LIS2DS12	16	0.5 - 3200	32,768	±2g, ±4g, ±8g, ±16g	120	150	1.62 -1.98	19.54
LSM9DS1	16	0 - 400	32,768	±2g, ±4g, ±8g, ±16g	N.A.	600	1.9 – 3.6	18.63
MMA8452	12	0 - 400	2048	±2g, ±4g, ±8g	126	165	1.95 – 3.6	10.49
MPU6050	16	5 - 260	32,768	±2g, ±4g, ±8g, ±16g	400	500	2.4 – 3.5	29.95

In selecting the accelerometer to be used by our sensor nodes, we focused on attributes such as sensitivity, noise floor, frequency of operation (bandwidth), current consumption, and cost. Besides, given that the amplitude of the surface vibrations on plastic pipes are of lower order magnitude and exist at low frequencies, the operation of the accelerometers in the ±2g is more suitable since it is more sensitive to small changes in acceleration compared to the other sensing ranges of the accelerometers. Hence, for accurate leak detection, we will need an accelerometer with high sensitivity and low noise floor level. From Table 3.4, we see that BMI160, LIS2DS12, LSM9DS1, and MPU6050 have very high sensitivities since they have a resolution of 16 bit. From

an energy consumption perspective, we see that LIS2DS12 provides the lowest current consumption, making it more suitable for a low-power solution. However, in terms of availability and cost, the LSM9DS1 was more available in the market. Thus, we selected the LSM9DS1 because of its low cost, high sensitivity, and availability. To reduce the power consumption of the sensor node, we also selected a low-power accelerometer to compensate for the high-power consumption of the more sensitive LSM9DS1 accelerometer by implementing hierarchical sensing. The low-power accelerometer is less sensitive and is continuously used to monitor the pipeline. Once it detects an acceleration larger than a defined threshold, it triggers an interrupt to wake up the more sensitive accelerometer from sleep to collect more accurate measurements that will be used for leak detection. From Table 3.4, we see that ADXL344, ADXL345, and ADXL362 have lower current consumption compared to the other accelerometers. ADXL362 has the lowest current consumption, but its sensitivity is low, which can compromise the accuracy of the leak detection system. Besides, the ADXL362 does not have a wide bandwidth (frequency of operation), unlike the ADXL344 that has a wide bandwidth that makes it suitable for detecting leaks on small diameter plastic pipes that are more visible at higher frequencies [153]. Thus, we selected ADXL344 because of its low-cost, low-power consumption, wide bandwidth, and finally its high sensitivity when compared to ADXL362. Another attractive property of the ADXL344 is its activity and inactivity sensing capability, which enables the sensor to generate an interrupt once the acceleration measured is above a predefined threshold value. This attribute is relevant in achieving low-power consumption, as the ultra-low power consumption and the threshold detection capability of the ADXL344 can permit continuous monitoring of the pipeline and only trigger a wake-up of the more sensitive accelerometer from sleep when an activity occurs (i.e., an acceleration greater than the predefined threshold value is detected). This can enable the achievement of our dual objectives of reliable real-time monitoring and low-power consumption.

Now that we have selected all the components that will make up our custom sensor node, we will, in the next section, fully describe the COTS components selected as constituent parts of the sensor node hardware and also discuss the sensor node design and configuration.

3.4 System Design

The requirement for a sensor node in a fully distributed WWPM system to be low-power and low-cost and have sufficient onboard computing resources led us to select cheap and low-power COTS components to design a low-cost and low-power custom sensor node (rather than using the research and commercial nodes available in the literature) to meet the specifications suited for our fully distributed solution. This was to achieve our dual objective of producing a low-cost and low-power sensor node for WWPM since we are aware that the choice of sensor node components is crucial in determining the cost, performance, and power consumption of the sensor node. Thus, in this section, we describe the specifications of the COTS components used to design the custom sensor node.

3.4.1 Sensor Node Architecture

Our proposed sensor node consists of an ESP32 from Espressif Systems as the processing unit, an nRF24L01+ transceiver module from Nordic as the communication unit and an LSM9DS1 accelerometer from STMicroelectronics and an ADXL344 accelerometer from Analog Devices as the sensing unit.

3.4.1.1 ESP32

ESP32 is a low-cost, low-power SoC increasingly used in the hobby and research development of IoT systems. This chip is widely used in tiny devices embedding Python or its derivatives (MicroPython, CircuitPython, etc.) for wireless embedded systems driven by a strong community such as Pycom [154] or CircuitPython [155]. This chip, although quite unused in traditional WSN hardware [4] as a result of it being a middle-end device [88], [89], has two main advantages: a 32 bit dual-core unit and an ULP coprocessor for low computation tasks. Last but not the least, it presents a wide support for conventional Operating Systems but also for more prospective Real-Time Operating Systems such as RIOT [156], Zephyr or Zerynth [157], FreeRTOS [88], etc.

From a technical point of view, the ESP32 offers:

- for computation: an Xtensa Dual-Core 32-bit LX6 microprocessor operating at up to 240 MHz, a 520 kB Static Random-Access Memory (SRAM), a 4 MB flash memory

- for interfacing: a 12-bit ADC with up to 18 channels and 40 physical General Purpose Input Output (GPIO) pads, which can be used as general purpose I/O to connect new sensors, or can be connected to an internal peripheral signal [15].
- for communication: a built-in Wi-Fi card supporting IEEE 802.11 b/g/n standards, Bluetooth version 4.2 and 486 BLE. Dedicated RF transceivers (such as nRF24L01+) can be added through GPIO to extend the RF physical layer support of ESP32 to IEEE802.15.4 protocols commonly used in the WSN community.

Figure 3.4 displays the functional block diagram of the ESP32 with all the features listed above and Table 3.5 presents a summary of the features of the ESP32.

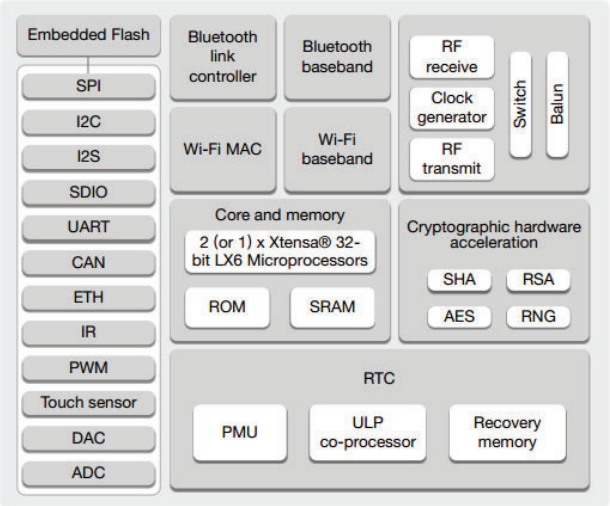

Figure 3. 4: Functional block diagram of the ESP32 **[158]**

Engineered for mobile devices, wearable electronics, and IoT applications, the ESP32 offers advanced power management features such as ULP consumption through power saving features including fine-resolution clock gating, multiple power modes, and power scaling [15]. The ESP32, when active (with the modem being off and CPU being operational), consumes currents in the [20 mA–68 mA] range and in the [10 µA–150 µA] while performing in the ULP state (only the RTC memory, RTC peripherals and the ULP co-processor are functional).

Table 3. 5: Summary of features of the ESP32

	Chip		
Features	**ESP32**		
Logic/DSP			
Processor	Tensilica Xtensa LX6 32 bit Dual-Core at 80/160/240 MHz		
DSP Block	32-bit multiplier 32-bit divider 40-bit MACC		
Memory			
SRAM	520 KB		
Flash	2 MB (max. 16 MB)		
Connectivity			
Wi-Fi	802.11 b/g/n		
Bluetooth	4.2 BR/EDR + BLE		
UART	3		
SPI	4		
I²C Interface	2		
I²S Interface	2		
Input/Output			
GPIO	32		
PWM	8		
ADC	18 (12-bit)		
DAC	2 (8-bit)		
Power consumption			
Operating Current	Modem Active	95 mA – 240 mA	
	Modem Sleep (main CPU powered on)	@ 240 MHz CPU speed	30 mA ~ 68 mA
		@ 160 MHz CPU speed	27 mA ~ 44 mA
		@ 80 MHz CPU speed	20 mA ~ 31 mA
	Light sleep	0.8 mA	
	Deep sleep (ULP active)	10 µA - 150 µA	
Operating Voltage	2.2 – 3.6 V (typical 3.3 V)		

From Table 3.5, we see that the ESP32 has enough onboard computational resources that can be used for in-situ processing, thus making it a good candidate for the processing units of sensor nodes that need to implement distributed computing. However, the power consumption is high, which makes it not very suitable for battery-powered WSN applications. But this power-hungry nature of the ESP32 chip can be taken care off by putting the main CPU at sleep most of the time while using the ULP coprocessor of the ESP32 as the control unit of the sensor node. The main CPU is only woken from deep sleep when there is powerful processing that needs to performed. By activating the ULP coprocessor and putting the main CPU of the ESP32 into deep most of the times, we can have a node which is computationally powerful, but consumes less energy. By so doing, we can achieve real-time monitoring and low-power consumption. Finally, there are several

development boards available for the ESP32 module. However, we used the ESP32 board produced by Adafruit called the Adafruit feather Huzzah32.

3.4.1.2 nRF24L01+

The CCXXXX (i.e., CC1000—the first generation of WSN, CC2420—the majority of WSN nodes developed in 2000s and 2010s, and CC2520—the new generation of WSN nodes) transceiver series from Texas Instruments are the most commonly used communication units in Wireless Sensor Nodes hardware development. However, we decided to use the nRF24L01+ due to its low power consumption on the one hand and for its burst mode (increased data rate) on the other hand. Even though the nRF24L01+ does not directly implement a mesh network at the MAC layer, the single dimensional aspect of WWPM systems reduces this drawback significantly.

At the physical layer, nRF24L01+ implements Gaussian Frequency Shift Keying (GFSK) modulation, with data rates ranging from 250 Kbps to 2 Mbps. A communication range of nearly 100 m and 500 m can be achieved with and without an external antenna, respectively, at maximum power [159], [160]. It is the perfect complementary RF transceiver for our node since it covers a longer range compared to Bluetooth, consumes less power than Wi-Fi and it is quite cheap from a financial point of view. Table 3.6 depicts the properties of the nRF24L01+ transceiver.

Table 3. 6: Summary of features of the nRF24L01+

	Chip		
Features	**nRF24L01+**		
Data Rate	250 kbps – 2 Mbps		
Operational frequency band	2.4 GHz		
Communication interface	SPI		
Modulation	GFSK		
Protocol	ANT		
Transmission range	100 m (up to 500 m with external antenna)		
Power consumption			
Operating current	Transmitter	@ 0 dBm output power	11.3 mA
		@ -6 dBm output power	9.0 mA
		@ -12dBm output power	7.5 mA
	Receiver	@ 2 Mbps	13.5 mA
		@ 1 Mbps	13.1 mA
		@ 250 kbps	12.6 mA
	Power down	900 nA	
	Standby-I mode	26 µA	
	Standby-II mode	320 µA	
Voltage Supply	1.9 – 3.6 V (typical 3.3 V)		

Furthermore, the embedded baseband protocol engine (Enhanced ShockBurst) implemented by the nRF24L01+ transceiver permits it to achieve higher data rates and can help to reduce the power consumption of the sensor node. This burst mode is attractive for distributed computing in WSN since it allows the transceiver to be active only for a short period during transmission. Besides, the nRF24L01+ also has a ULP power-down state which consumes current of the order of 0.9 μA, which can be used to reduce the power consumption incurred via idle listening of the nRF24L01+ transceiver. Thus, rather than leaving the nRF24L01+ in the idle listening mode which consumes current in the order of 13 mA, the transceiver can be placed in the standby mode when there is no activity and will be awakened from sleep by an externally-controlled wake-up, which in our case can be an environmentally controlled wake-up such as interrupt from the accelerometer when the measured acceleration exceeds the predefined threshold. However, care must be taken in placing the transceiver in the power down mode to prevent the loss of packets from neighbouring sensor nodes. This can be attained by waking the transceivers of all neighbouring nodes once an activity has been detected by the accelerometer, i.e., when the measured acceleration is above the predefined threshold.

A number of nRF24L01+ boards exist in the market. However, we made use of the nRF24L01+ board from Sparkfun.

3.4.1.3 LSM9DS1

LSM9DS1 is a 9 Degrees of Freedom (DOF) IMU which features a 3D digital linear acceleration sensor, a 3D digital angular rate sensor, and a 3D digital magnetic sensor. Additionally, the LSM9DS1 has a linear acceleration full scale of ±2 g/±4 g/±8 g/±16 g, a magnetic field full scale of ±4/±8/±12/±16 Gauss and an angular rate of ±245/±500/±2000 dps (degree per second). It also includes an I^2C serial bus and an SPI serial standard interface for interfacing with the MCU. It has an analogue supply voltage ranging from 1.9 V to 3.6 V and provides an ultra-low current consumption of 600 μA when the accelerometer is in the normal mode [161]. It has three 16-bit ADCs for digitizing the accelerometer outputs which can result in more accurate digital outputs and has a wide accelerometer sensing range for tracking both slow and fast motions. In addition to the low-power consumption feature of the LSM9DS1 IMU, another interesting feature of this sensor which makes it relevant to achieving low-power consumption is the ability of the sensor to

generate an interrupt once the acceleration measured is above a predefined threshold value. With this property, the ESP32 can be programmed to operate in deep sleep mode most of the time. The LSM9DS1 will continuously monitor the pipes at all times for any deviation from the predefined threshold value. Upon deviation detection, the sensor sends an external wake-up interrupt signal to start the ESP32 main core. Thus, this threshold detection property of the LSM9DS1 makes it very useful in providing a low-power solution, as it can be used to reduce the power consumption of the sensor node and extend the lifespan of the WSN. It is also low cost. The features of the LSM9DS1 accelerometer are summarised in Table 3.7.

Table 3. 7: Summary of features of the LSM9DS1

	Chip			
Features	**LSM9DS1**			
Resolution	16 bits			
Bandwidth	0 – 400 Hz			
Output Data Rate	10 – 952 Hz			
Communication interface	SPI and I2C			
Zero-g level offset	X-axis	Y-axis	Z-axis	
	0 g	0 g	1 g	
Zero-g level offset accuracy	±90 mg			
Sensing range	±2g, ±4g, ±8g, and ±16g,			
Sensitivity (LSB/g)	±2g	±4g	±8g	±16g
	16384	8192	4096	2048
Power consumption				
Operating current	600 µA			
Voltage Supply	1.9 – 3.6 V			

A number of LSM9DS1 boards exists in the market. However, we made use of the LSM9DS1 evaluation board from ST Microelectronics.

3.4.1.4 ADXL344

The ADXL344 sensor of Analog Devices is a MEMS three-axis accelerometer with a selectable sensing range (±2g, ±4g, ±8g, and ±16g) and bandwidth (0 – 1600 Hz) and configurable, built-in motion detection which makes it suitable for sensing acceleration in a wide variety of applications. The ADXL344 functions using a capacitive transduction mechanism and it is supplied in a small,

thin, 3 mm × 3 mm × 0.95 mm, 16-terminal, plastic package. Table 3.8 summarizes the properties of the ADXL344 accelerometer.

Table 3. 8: Summary of features of the ADXL344

	Chip			
Features	**ADXL344**			
Resolution	13 bits			
Bandwidth	0 – 1600 Hz			
Output Data Rate	0.1 – 3200 Hz			
Communication interface	SPI and I2C			
Zero-g level offset	X-axis	Y-axis	Z-axis	
	0 g	0 g	1 g	
Noise floor level (µg/√Hz)	X-axis	Y-axis	Z-axis	
	420	420	530	
Zero-g level offset accuracy	±35 mg			
Sensing range	±2g, ±4g, ±8g, and ±16g,			
Sensitivity (LSB/g)	±2g	±4g	±8g	±16g
	2048	1024	512	256
Power consumption				
Operating current	23 µA			
Voltage Supply	1.7 – 2.75 V			

The ADXL344 has a resolution of 13 bits and can have a sensitivity as high as 2048 LSB/g. The measured acceleration from the ADXL344 can be sent to the interfacing MCU by means of SPI or I2C communication, since ADXL344 possesses both interfaces. The sensing range of the ADXL344 is user-programmable with options such as ±2g, ±4g, ±8g, and ±16g available. The ADXL344 also has a wide frequency response (bandwidth) ranging from 0 – 1.6 kHz and an output data rate (sampling frequency) ranging from 0.1 – 3.2 kHz. In terms of current consumption, the ADXL344 is an ultra-low power device consuming current as low as 23 µA. It also has a voltage supply range from 1.7 – 2.75 V. In addition, the activity and inactivity sensing to detect the presence or lack of motion are also special sensing functions provided by the ADXL344 accelerometer. This activity and inactivity sensing capability of the ADXL344 can be used to achieve low power consumption for our custom sensor node by putting all the other components (ESP32 microcontroller, nRF24L01+ radio module, and LSM9DS1 accelerometer) to sleep and leaving only the ADXL344 (which consumes less power) active. That is, the sensor node goes to sleep when there is no activity (when the acceleration measured by the ADXL344 is below a

certain threshold) and a trigger from the ADXL344 when there is an activity will wake-up the sensor node from sleep. Finally, when the ADXL344 is taking measurements, it consumes a current of 140 µA while when it is not taking measurements, it is in the standby mode which consumes a current of 0.2 µA.

In the next section, we will describe how we configured the hardware components of our custom sensor node.

3.5 Configuration of the Sensor Node

The nRF24L01+ transceiver module is interfaced with the ESP32 via the SPI interface, whereas the LSM9DS1 and ADXL344 sensors are interfaced with the ESP32 via the I^2C interface. Figure 3.5 represents the interconnection of the sensor node's components.

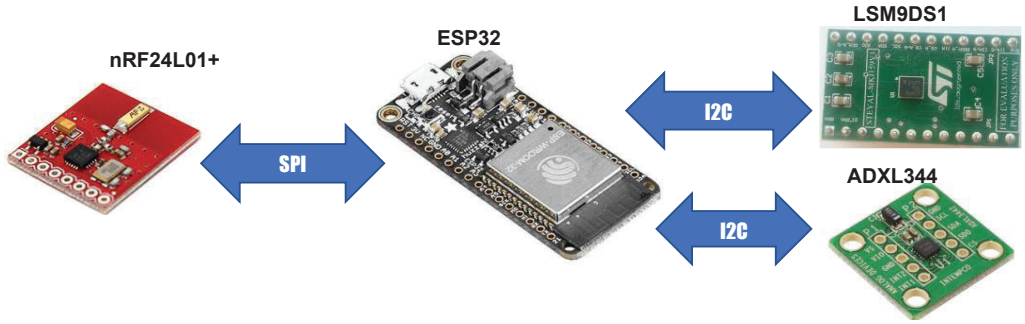

Figure 3. 5: Hardware interfacing of the sensor node's components.

3.5.1 Configuration of LSM9DS1 and ADXL344

Since we are interested in measuring vibrations of low magnitude, the sensitivity of the LSM9DS1 and ADXL362 accelerometer sensors are configured to ±2 g since this has the highest sensitivity (0.061 mg/LSB for LSM9DS1 and 3.9 mg/LSB for ADXL344), which makes it most appropriate for detecting vibrations of smaller magnitudes such as those on the surface of a water pipe.

As we saw in sub-section 2.6.2 of Chapter 2, the reliability of leak detection in plastic pipes is dependent on the pipe size, interfering noise, and frequency response of the accelerometer. Thus, for reliable leak detection, it is required that the frequency response of the accelerometer accommodates the bandwidth in which leak noise is found. In a study carried out by Scussel et al.

[153], the results showed that when using accelerometers to detect leaks in small diameter plastic pipes, the leak noise is in a much higher frequency than in the case of large diameter pipes. Their results revealed that for a 35.8 mm plastic pipe, the leak noise is found in the frequency band 273 – 746 Hz. This is because the measurements at lower frequencies are dominated by noise from low-frequency vibrations, e.g., noise from pumps. Based on this, we configured the bandwidth of the ADXL344 to 900 Hz and the sampling frequency to 1800 Hz, while we configured the bandwidth and sampling frequency for the LSM9DS1 to 400 Hz and 953 Hz, respectively.

In addition, both accelerometers measure the vibration in 3D, i.e., in the X, Y, and Z directions given by Ax, Ay, and Az, respectively. The actual acceleration in each axis is computed by subtracting the zero-g offset from the measured acceleration in that axis and then the overall magnitude of the pipe surface acceleration is obtained by taking the resultant acceleration in all three directions. Lastly, when both accelerometers are used, the LSM9DS1 is always placed in the power-down mode and only gets to the normal mode once a wake-up is triggered by the ADXL344 when an activity is detected. A predefined threshold value is set in the activity threshold register of the ADXL344 after leak characterization and it is used to determine the occurrence or absence of activity (leak).

3.5.2 Configuration of nRF24L01+

The transmitter of the nRF24L01+ radio module was configured to transmit at -12 dBm output power for two reasons. The first reason was to achieve low-power consumption since the transmitter consumes a current of 7.5 mA when transmitting at -12 dBm output power. The second reason was that at -12 dBm the transmission range of the transmitter is shorter. This is good for our fully distributed WWPM solution since nodes only communicate with their close neighbours and the internode spacing is short since leak vibrations do not go far in plastic pipes and because MEMS accelerometers were used to monitor the pipe surface vibrations. The receiver of the nRF24L01+ radio module was configured to operate at 2 Mbps with a sensitivity of –82 dBm. This was in order to reduce the amount of time the transceiver is active during reception and transmission. Another reason was to reduce the number of collisions between sensor nodes since they use the same radio channel, and the nRF24L01+ does not explicitly implement any MAC algorithm for channel access. To reduce packet loss resulting from packet collision, it is important

to implement retransmission and stagger the transmission intervals assigned to neighbouring nodes. Thus, the nRF24L01+ was configured to perform 15 retries every after 500 µs when a failed transmission occurred. In addition, to prevent the loss of packets resulting from the RX FIFO of the transceiver being full, the interrupt pin (IRQ) of the nRF24L01+ was configured to generate an external interrupt to the ESP32 each time a packet was received. This triggered the immediate reading of the packet from the RX FIFO.

Finally, to provide power-efficient listening when there is no activity, it would be useful for nodes listening to sleep for extended periods if they know that they would not miss packets. Thus, the radios of the nodes were configured to go to sleep when there is no activity and only awaken when there is an activity i.e., when a trigger is received from the ADXL344 sensor. The idea is that the occurrence of a leak will cause the acceleration measured by the ADXL344 sensor to exceed the predefined threshold value stored in its activity threshold register. This will then generate an interrupt that serves as an environmentally controlled wake-up to a sensor node and its close neighbours (nodes found in the vicinity where the leak occurs). By so doing, low-power consumption can be achieved without a reduction in the reliability of the system.

3.5.3 Configuration of ESP32

The ESP32 was configured to operate in the modem sleep mode when there is an activity and in the deep sleep mode when there is no activity. The modem sleep mode rather than the modem active mode was used since we are not using the Wi-Fi or BLE module of the ESP32 for communication. The CPU clock speed of the ESP32 was set to 80 MHz (with a datasheet current consumption of 20 mA ~ 31 mA) since it is sufficient run the distributed algorithms that will be implemented on the sensor node and also consumes less energy compared to a CPU speed of 160 MHz or 240 MHz. To reduce the node's power consumption when there is no activity, it is advantageous to harness the ULP coprocessor of the ESP32 by putting the node into deep sleep mode (with current consumption in the range 10 µA –150 µA). Thus, the node will only be active for very short periods of time, i.e., when it needs to transmit data and perform some processing on data received from neighbouring sensor nodes. When there is no activity, the ESP32 goes into the deep sleep mode where the main CPU core is inactive and the ULP coprocessor is active and is used for controlling the sensor node peripherals. The ESP32 was configured to wake-up from deep

sleep by means of an external interrupt, which is the interrupt generated by the ADXL344 sensor when the measured acceleration exceeds the predefined threshold stored in the activity threshold register of the ADXL344 sensor. It should also be noted that in this study we did not use any real-time operating system on the ESP32. The programs implementing the algorithms were written in C/C++ using the Arduino IDE since we were dealing with simple algorithms in this study. However, a real-time operating will be handy in future work.

Now that we have completed the selection of the sensor node components, designed the node, and configured it, we will in the following sections discuss the design and configuration of a custom power measurement device. The purpose of this measurement device is to measure the power consumption of the sensor nodes in the laboratory experimental setup. It is important to measure the energy consumption of our proposed solution because minimizing the sensor node's energy consumption and maximising WSN lifetime (which is one of the objectives of this thesis) is a major challenge in WWPM.

3.6 Design of a Remote Power Measurement Device

In this section, we describe the development of a power measurement device we call a USB power meter that we used to measure the power consumption of the sensor node in order to establish the energy profile of our distributed solution. This work was carried out by Khouloud Amira, a second-year master's student on an internship in Ampere Laboratory, under my guidance. We developed this device for two main reasons: (1) to enable us to measure very low currents in the µA range particularly current consumption of the node when the ESP32 is operating in deep sleep mode and (2) to be able to monitor and store the power consumption of the nodes without physically being present (i.e., recording of power measurements collected periodically over a long period of time).

Current consumption monitoring is a very important aspect for battery-powered sensor nodes since they are constrained in terms of energy. Selecting the correct method to monitor the current consumption of a sensor node is critical in optimizing the system performance. There are three primary approaches used to profile the power of systems and components. They include simulator-based power estimation, direct measurements, and event-based estimation [162]. We made use of the direct measurement method which can be done via either operational/difference (milliamps to

tens of amps), instrumentation (nanoamps to tens of amps), or current sense (tens of microamps to tens of amps) methods, and where power can be directly measured both intrusively or nonintrusively. The intrusive measurements require inserting precision (shunt) resistors into the power supply lines of components under study and use power meters to measure the voltage drop across the resistor. The current through the component is calculated by the voltage drop over the shunt resistor divided by its resistance. The nonintrusive approach uses ammeters to measure the current flow of the power supply lines directly [162]. Since the currents we were measuring were in the range of tens of microamps to amps, we used the INA226 device that employs the current sense method. Hence, our custom power measurement device was based on the intrusive direct measurement method and was composed of a 100 mΩ shunt resistor, INA226 module, STM nucleo-32 F303k8 microcontroller, nRF24L01+ transceiver, 128 × 64 OLED display, SD card and two USB ports. Figure 3.6 is a block diagram display of the USB power meter.

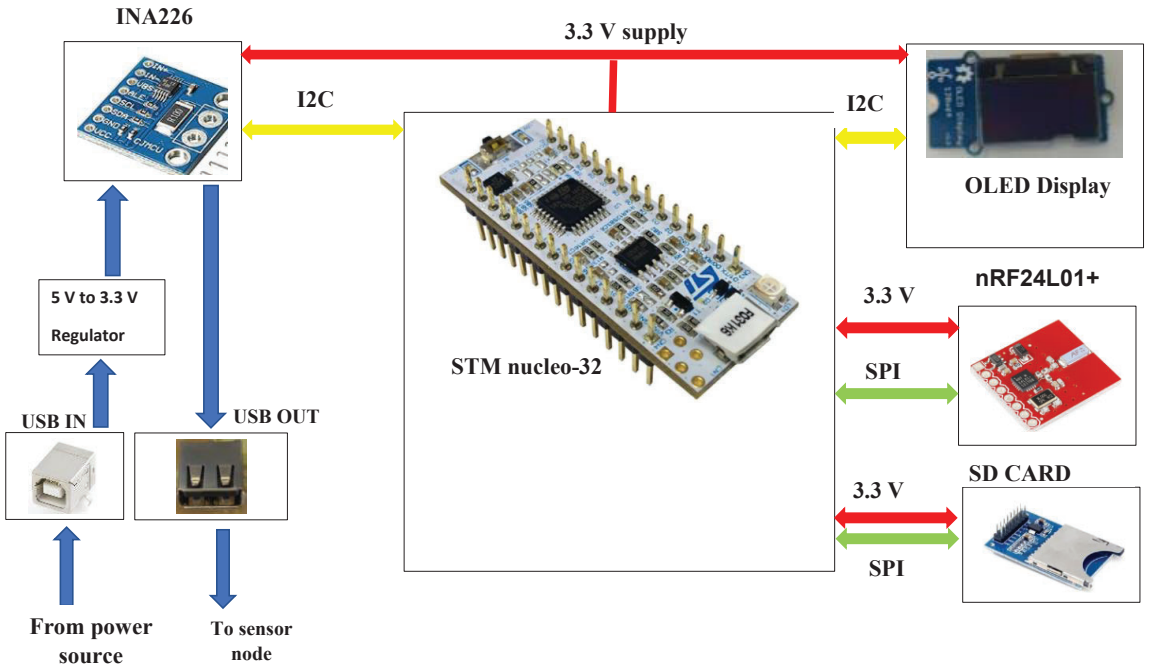

Figure 3. 6: Block diagram of power measurement device

The INA226, often used in instrumentation for low current monitoring, is a current shunt and power monitor with an I^2C compatible interface. It was used to monitor both the shunt voltage drop across the shunt resistor (placed in series with the sensor node) and the bus supply voltage.

The INA226 can be used either in high-side sensing (where a shunt resistor is placed between the supply voltage and the load) or low-side sensing (where a shunt resistor is placed between the load and the system ground). High-side sensing was used since it is preferable when dealing with low currents given that it is more responsive to changes in the current flow and it adds no disturbance to system ground [163]. In addition, the INA226 is designed for a maximum input shunt voltage of 81.92 mV and has a 16-bit ADC. Thus, the maximum current that can be measured by the device is 819.2 mA and the resolution is 25 µA in the case where a 100 mΩ shunt resistor is used. Using the 100 mΩ shunt resistor enabled us to measure the current consumption of the ESP32 when it is in the deep sleep mode (with currents in the µA range). Besides, the voltage drop across the shunt resistor allowed sufficient voltage to power the ESP32. We used the STM nucleo-32 F303k8 MCU to interface with the INA226 and also to configure its programmable calibration value, conversion time and averaging mode. For better accuracy in measurements, we configured the conversion time and averaging mode to 140 µs and 4, respectively. The OLED display was used for displaying the power measurements, the SD card for storing the power measurements over a long period of time and the nRF24L01+ transceiver permitted the remote reading of the power consumption. USB-IN was used to supply power to the measurement board while USB-OUT supplied the sensor node.

3.7 Summary

To provide a reliable, real-time, fully distributed, and low-power WWPM solution for monitoring plastic WDNs in developing countries, it is necessary that the sensor nodes be cheap and also consume less energy (since the nodes are to be powered by a battery and also because the WWPM is required to go unattended for a long time without replacing the battery). This led us to perform a thorough review of existing commercial and research sensor nodes by focusing on their computing power, energy consumption, and cost. The need for a sensor node that is low-cost and has a high computing power but consumes low power led us to design a custom node rather than using existing research or commercial sensor nodes. The custom node designed consisted of an ESP32 SoC as the processing unit and an nRF24L01+ transceiver module as the communication unit. The ESP32 was chosen because of its low-cost, high computing power (since it incorporates a dual-core 32-bit Xtensa LX6 processor with a processing speed of up to 240 MHz as the main

core), and low-power consumption (since it incorporates a ULP coprocessor with current consumption in the range 10 μA – 150 μA). The high computing power of the ESP32 main core permits our custom node to be capable of local processing while the ULP coprocessor permits it to achieve real-time monitoring while maintaining low energy consumption. This meets our objectives of a real-time, fully-distributed, and low-power WWPM solution. The nRF24L01+ transceiver module was selected because of its low-cost, burst mode, and low-power consumption (with TX/RX peak currents less than 14 mA). To achieve reliable leak detection on plastic pipes while maintaining low energy consumption, we proposed the implementation hierarchical sensing by selecting both a low accuracy (but low-power accelerometer) and a high accuracy accelerometer. The accelerometers selected consist of the ADXL344, which was chosen because of its low-power consumption and event detection capability, and the LSM9DS1 which was chosen because of its high sensitivity and low-power consumption. In this chapter, we also discussed the configuration of the sensor node's component needed to achieve distributed computing, real-time monitoring, and low-power consumption. We ended the chapter by discussing the design and configuration of a custom power measurement device that will be used for measuring the power consumption of the sensor node.

Chapter 4

Distributed Kalman Filter for Wireless Sensor Networks

The purpose of this study is to investigate and evaluate the implementation of a reliable, real-time, fully distributed, low-cost and low-power WWPM solution for leak detection in plastic WDN. We aim to achieve:

1. cost-effectiveness by using low-cost COTS components as the building blocks of the sensor node,
2. low-power consumption by implementing duty cycling and hierarchical sensing on the sensor node, and
3. reliable and real-time leak detection by implementing distributed computing within the WWPM system.

In this chapter, the objective is to select computationally less intensive distributed Kalman filter (DKF) algorithms to be implemented within the sensor nodes to enable reliable real-time leak detection while preserving the lifespan of the WWPM system. The challenge is that there exist numerous DKF algorithms in the literature with different performances, complexities, computational requirements, etc. Thus, it is necessary to select DKF algorithms that are suitable for WSN applications. The goal of this chapter is to get us close to the realization of point 3 above.

We start this chapter by presenting the Kalman Filter (KF) with reasons why we selected it as the signal processing algorithm for improving the quality of pipe surface vibration measurements collected using low-cost MEMS accelerometers. We later review the categories of DKF algorithms for low-cost WSN applications such as WWPM systems. We then select three DKF algorithms that we will implement and evaluate their performance and power consumption via simulations and physical experiments. We end the chapter by presenting the metrics to evaluate the performance and power consumption of our proposed fully distributed leak detection solution and provide a layout of the experiments that will be conducted in chapters 5 and 6.

4.1 The Standard Kalman Filter and Data Fusion

4.1.1 Reasons for choosing Kalman Filter

As mentioned in sub-section 2.6.3, there are several signal processing techniques for processing leak data collected from plastic pipes using low-cost MEMS accelerometers. However, in this study, we used KF (which is undoubtedly one of the most popular data fusion technology used in WSN applications).

The reasons for choosing KF as the signal processing technique for increasing the reliability of our proposed leak detection solution include:

1. It requires less memory given that it does not need to store any previous history other than the previous state. This makes it suitable for implementation in sensor nodes which are constrained in memory, unlike other digital filtering algorithms which require taking a sliding window of noisy data in order to perform the filtering function.
2. It can achieve rapid detection of events (e.g., leaks) as they occur in real-time. The fast-processing capability of KF stems from the fact that it straightforwardly processes data in the time domain, making it a suitable choice for implementation in real-time embedded applications. With one of the objectives of this study being to provide a real-time solution for leak detection in WWPM, KF is a suitable choice for filtering noisy leak data because it is fast in processing time series data.
3. It is also computationally less intensive, making it suitable for implementation in sensor nodes constrained in terms of computing power. Additionally, this particular feature of KF will reduce the energy consumed by the node when processing leak signals and thus extend the lifetime of the sensor nodes.
4. There is a linear relationship between the measurement and the interested state, given that the state of interest (pipe surface acceleration) can be measured directly using the MEMS accelerometers. Therefore, KF applicable to linear dynamic systems can thus be utilized in our proposed WWPM solution.

4.1.2 Description of the Standard Kalman Filter

As earlier stated in section 1.6, KF can be applied as a state observer or sensor fusion algorithm. In this study, we focus on applying KF as a sensor fusion algorithm [164]. The reason is that our proposed WWPM is categorized as a signal processing-based technique (Figure 2.1 of chapter 2), and like other WWPM techniques based on signal processing [13], [38], [50], it involves processing vibration data collected from low-cost MEMS accelerometers mounted on the pipe surface in order to detect leaks. In this case, the state of interest (pipe surface acceleration) is directly measured using the MEMS accelerometers. This is unlike the model-based methods (e.g., state observers) that need an accurate theoretical model of the pipeline dynamics to detect leaks since the state of interest (leak size, leak location, etc.) cannot be directly measured by the sensors [44], [101], [165].

Now, let's consider the linear stochastic discrete-time system below:

$$x_k = Ax_{k-1} + Bu_k + q_k \tag{4.1}$$

$$y_k = Hx_k + r_k \tag{4.2}$$

where x_k and x_{k-1} represent the state of the system at time k and k-1, respectively. *A* and *B* are matrices defining the system dynamics, u_k is the input vector, *H* is a matrix defining the relationship between the measurement and the interested state, y_k is the measurement at time k, q_k and r_k are the process noise and measurement noise, respectively.

Applying KF as a sensor fusion algorithm to the linear stochastic discrete-time system described in equations (4.1) and (4.2) involves two steps: prediction (time update) and correction (measurement update). The prediction step is followed by a correction step to determine the states of the filter. This is sometimes called predictor–corrector or prediction–update [166].

In the first step, the state of interest, x at time k, is predicted from the updated state at time k-1. The prediction of the current state and the state error covariance matrix are given by:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k \tag{4.3}$$

$$P_k^- = AP_{k-1}A^T + Q_k \tag{4.4}$$

where $\hat{x}_k^-$ is the predicted state vector at time k, $\hat{x}_{k-1}$ is the previous estimated state vector, P_k^- represents the predicted state error covariance matrix, P_{k-1} is the previous estimated state error covariance matrix, and Q_k is the covariance matrix of the process noise q_k.

The second step is the correction or update step. This step aims to get an improved estimate by incorporating new measurements into the predicted estimate using the Kalman gain (K_k). KF at this step actually performs data fusion between predicted state and the measurements to obtain an optimal state estimate.

$$K_k = P_k^- H^T (H P_k^- H^T + R_k)^{-1} \tag{4.5}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (y_k - H\hat{x}_k^-) \tag{4.6}$$

$$P_k = (I - K_k H) P_k^- \tag{4.7}$$

where R_k is the measurement noise covariance, I is an identity matrix, $\hat{x}_k$ is the estimated or updated state vector, and P_k is the updated state error covariance matrix.

The equations (4.3) through (4.7) represent what is generally referred to as the standard KF. However, in this study we call it Local Kalman Filter (LKF) when it is implemented at the sensor node level to emphasize on the fact it is locally processed using the onboard computing resources of the sensor node. By actually performing KF using the onboard computing resources of each sensor node in the WWPM system, the LKF algorithm actually implements distributed computing within the WSN. However, the local estimates obtained at each node are not fused with those of other nodes given that the nodes in the LKF implementation do not communicate with each other. It should be recalled that the study conducted by Karray et al. [38] implemented an LKF, even though they used a force sensitive resistor rather than a low-cost MEMS accelerometer for pipe surface vibration measurements.

Now that we have given reasons why KF was selected for use in this study and briefly described the KF technology and how it can implement data fusion at the node level, we will in the next section present a review of KF for WSNs. We will start by reviewing a number of KF algorithms that can be implemented in WSNs, and finally select and detailly describe three KF algorithms that we will implement in this study.

4.2 Application of Kalman Filter for Multi-sensor Data Fusion in Low-cost WSNs

The implementation of KF in multi-sensor systems such as a WSN differs from the implementation of KF in single sensor systems. The implementation of KF in a WSN application can either be via a Centralized Kalman Filter (CKF) or Distributed Kalman Filter (DKF). CKF is a KF employing centralized data fusion, whereas DKF is a KF with distributed data fusion. In centralized data fusion, all the sensor nodes send their measurements to a central node, which fuses them to obtain an accurate global estimate. However, in distributed data fusion, the sensor nodes merge their estimates with those of their close neighbours via local communications that exist between them to obtain an accurate estimate rather than transmitting to a central site for centralized fusion. In the next sub-sections, we present a brief description of CKF alongside its drawbacks and how these drawbacks are handled by DKF. We then provide additional advantages of DKF that resulted in its selection and implementation in this study.

4.2.1 Centralized Kalman Filter

CKF is Bayesian optimal since it fuses data from all the sensor nodes. However, in large-scale WSN applications, it is difficult to fuse data from all the sensor nodes at a single fusion centre. For this reason, CKF is usually never implemented in practice, but it is used as a benchmark for evaluating the performance of distributed and decentralized KF algorithms [59].

Now, let's consider a discrete-time linear-stochastic and dynamic system which is being monitored by N sensor nodes described by the mathematical formulation below:

$$x_k = Ax_{k-1} + q_k \tag{4.8}$$

$$y_k^i = H_k^i x_k + r_k^i, \ \text{i} = 1, 2, 3, \ldots, \text{N} \tag{4.9}$$

where the measurement noise from two different sensors, r_k^i and r_k^j maybe cross-correlated at the same time step k. However, for simplicity, it is assumed that the measurement noise from different sensor nodes are uncorrelated [59].

At the central fusion centre, all the measurements from the N sensors at time step k can be expressed as:

$$y_k = H_k x_k + r_k \tag{4.10}$$

and the centralized estimate of the state x_k, that takes into consideration sensor data from all N sensor nodes can be obtained using the equations of the standard KF (equations (4.3) through (4.7)). This results in an updated state estimate represented in the information form of the KF as [60]:

$$(P_k)^{-1}\hat{x}_k = (P_k^-)^{-1}\hat{x}_k^- + \sum_{i=1}^{N} {H_k^i}^T (R_k^i)^{-1} y_k^i \tag{4.11}$$

where, the updated state error covariance of CKF is given by:

$$(P_k)^{-1} = (P_k^-)^{-1} + \sum_{i=1}^{N} {H_k^i}^T (R_k^i)^{-1} H_k^i \tag{4.12}$$

As can be seen from equations 4.11 and 4.12, CKF requires all the sensor nodes to send their measurements to the central fusion centre. However, this is practically impossible for large scale WSNs as some sensor nodes may fail to send their estimates to the central fusion centre at certain times due to the unreliable nature of the communication links of WSNs. CKF is therefore not suitable for implementation in our context of WWPM because of the drawbacks of centralized monitoring in WSN applications which we mentioned in section 2.2 of chapter 2. Drawbacks such as uneven distribution of energy leading to the energy hole effect, fast depletion of sensor node's energy due to frequent communication, reliability issues due to the unstable nature of wireless links, lack of robustness due to the presence of a single point of failure, high bandwidth requirement, and increase latency which undermines real-time applicability, are some of the disadvantages of centralized computing in WSN which is exploited by CKF. Due to these disadvantages, we switch to distributed computing by implementing a DKF instead. The drawbacks of CKF can be reduced by the distributed data fusion strategy employed in DKF algorithms. The aim is to perform distributed state estimation such that each sensor node is capable of obtaining an accurate estimate that is close to that obtained via centralized fusion [167] but at low computation and communication cost [64].

4.2.2 Distributed Kalman Filter

Unlike CKF where there is only one fusion centre for fusing the measurements from all sensor nodes, in DKF, each sensor node acts as a local fusion centre for fusing its own measurements with those of its close neighbours to achieve an accurate estimate which is close to that obtained by the theoretical Bayesian optimal CKF algorithm. This makes DKF algorithms suitable for practical implementation in large-scale WSN applications.

DKF algorithms have been used extensively in low-cost WSN-based target tracking applications [59]. They can also be used in any application where it is required to improve the accuracy of a monitored parameter by using redundant information from multiple low-cost sensors in order to complement the limitations of a single sensor node. They can be extended to applications such as navigation systems, environmental and power system monitoring, autonomous robot systems, large-scale camera networks, wireless channel monitoring, structural health monitoring of civil structures etc.

Another novel application where DKF can also be applied is the real-time and reliable detection of leaks in WDNs monitored by WSNs. This is the main focus and contribution of this thesis, and to the best of our knowledge, this is the first work that uses WSNs consisting of sensor nodes with low-cost MEMS accelerometers and implements a DKF algorithm on each sensor node for leak detection. The goal is to improve the reliability of leak detection in real-time and provide a fully distributed solution that curbs the limitations of centralized solutions, i.e., high latency (due to multi-hop communication), scalability issues, and high-power consumption. In our proposed solution, each sensor node runs a local Kalman filter (LKF) to obtain an accurate local estimate from the local measurements, then later fuses it with those of its close neighbours to achieve a more accurate global estimate used for leak detection. In this way, our proposed solution is autonomous and does not need any central intelligence.

In addition to the reasons for choosing the KF algorithm mentioned above, DKF was chosen as the as the distributed algorithm for implementing distributed computing within the WWPM system for the following reasons:

1. Its prediction capability can be used to reduce the amount of data traffic by making predictions and inferences about the monitored parameter based on previous estimates and

measurements. The number of communications in the WSN can be reduced by using DKF because the measurements from neighbouring sensor nodes are correlated, and this can permit the DKF algorithm implemented on one sensor node to be able to predict the measurement of another sensor without necessarily communicating with the node. This reduces the number of data transmissions in the WSN and the power consumption of the sensor nodes and thus extends the WSN lifetime.

2. Its data fusion capability can fuse sensor readings from neighbouring sensor nodes, thereby increasing the reliability, accuracy, and confidence of the pipe surface acceleration obtained from multiple sensors. This feature can improve leak detection accuracy.
3. Its distributed nature, which permits only local communications (limited between neighbouring sensor nodes), can reduce data transmission rate and thus provide energy savings in the WSN since the sensor nodes do not send their measurements to a central fusion centre.
4. It is more robust and scalable due to the lack of a single point of failure, given that each sensor node acts as a local fusion centre.

In summary, DKF incorporates both data prediction (an energy conservation technique to extend the lifespan of the WWPM system) and distributed multi-sensor data fusion (to improve the accuracy of the leak detection system).

In the next section, we will review DKF algorithms for low-cost WSNs and select three DKF algorithms that will be implemented in this study.

4.3 A Review of DKF Algorithms for Low-cost WSNs

Several DKF algorithms are available in the literature as can be seen in reviews on DKF [59], [168], [169]. In Addition, the DKF algorithms are widely categorized in the literature as either consensus-based or diffusion-based. However, He et al. [59] recently reviewed DKF algorithms for low-cost sensor networks and broadly classified them as either sequential, consensus, gossip, or diffusion, based on how local sensor nodes communicate with their neighbours to perform data fusion. The authors also evaluated DKF algorithms in terms of criteria such as global convergence (ability to converge to the value of the Bayesian optimal CKF asymptotically or in finite time),

local consistency (ability to maintain a consensus in the estimates of neighbouring sensor nodes), and communication burden (number of communication rounds involved during fusion).

According to the study by [59], the sequential-based DKFs involve communication between two sensors at any point in time and the fusion is performed repeatedly and sequentially. Hence, this type of distributed data fusion in only possible in linear WSNs. Consensus-based DKFs require each sensor node to transmit its local information and also receive information from all its neighbours at every time step and consensus iteration. Each node implementing a gossip-based DKF transmits its local information and receives information from just a selected neighbour at every time step and gossip iteration while each sensor node implementing a diffusion-based DKF transmits its local information and receives information from all its neighbours at every time step and with only one communication iteration involved.

In the sub-sections that follow, we describe in details consensus-based, gossip-based, and diffusion-based distributed data fusion techniques.

4.3.1 Consensus-based DKF Algorithms

Consensus-based DKF algorithms achieve global convergence by performing fusion in a distributed manner through an average consensus approach. The global estimate obtained from the distributed data fusion closely matches that obtained by the theoretical Bayesian optimal CKF algorithm. The sensor nodes achieve this global estimate by exchanging local information with their close neighbours in between measurements. Each exchange that occurs in between measurements is referred to as a consensus iteration and the number of consensus iterations between measurements is defined by L. Examples of popular studies where the consensus approach of distributed data fusion is applied in WSNs include: [64], [170].

Now considering that the system described by equations 4.8 and 4.9 is monitored by a WSN consisting of N sensor nodes and implementing a distributed architecture with each sensor node i having C_i connected neighbours, the goal of consensus-based DKF algorithms is to compute the CKF terms s = $\sum_{i=1}^{N} {H_k^i}^T (R_k^i)^{-1} y_k^i$ and S = $\sum_{i=1}^{N} {H_k^i}^T (R_k^i)^{-1} H_k^i$ in a distributed manner using the average consensus protocol [59]. At any time step, k, the updated state estimate and the updated

state error covariance at node i, after L consensus iterations between time step k-1 and k are given by:

$$(P_k^i)^{-1}\hat{x}_k^i = (P_k^{i^-})^{-1}\hat{x}_k^{i^-} + \sum_{j=1}^{Ci} H_k^{j^T}(R_k^j)^{-1}y_k^j \tag{4.13}$$

$$(P_k^i)^{-1} = (P_k^{i^-})^{-1} + \sum_{j=1}^{Ci} H_k^{j^T}(R_k^j)^{-1}H_k^j \qquad j \in C_i \tag{4.14}$$

Now based on the type of information that is exchanged between neighbouring sensor nodes during consensus iterations, there are two variants of consensus-based DKF algorithms. There are those that perform consensus on measurement (CM) by exchanging $H_k^{i^T}(R_k^i)^{-1}y_k^i$ and $H_k^{i^T}(R_k^i)^{-1}H_k^i$ during each consensus iteration and those that perform consensus on information (CI) by exchanging $(P_k^i)^{-1}\hat{x}_k^i$ and $(P_k^i)^{-1}$ during each consensus iteration.

For consensus-based DKFs that employ CM, the correction step at each sensor node, i, at the end of the final consensus iteration is given by:

$$(P_k^i)^{-1}\hat{x}_k^i = (P_k^{i^-})^{-1}\hat{x}_k^{i^-} + \omega_k^i g_{k,L}^i \tag{4.15}$$

$$(P_k^i)^{-1} = (P_k^{i^-})^{-1} + \omega_k^i G_{k,L}^i \tag{4.16}$$

where $g_{k,L}^i = \sum_{j=1}^{Ci}\pi^{i,j}H_k^{j^T}(R_k^j)^{-1}y_k^j$ and $G_{k,L}^i = \sum_{j=1}^{Ci}\pi^{i,j}H_k^{j^T}(R_k^j)^{-1}H_k^j$, with $\sum_{j=1}^{Ci}\pi^{i,j} = 1$. ω_k^i is used for compensating for the approximation of $g_{k,L}^i$ and $G_{k,L}^i$

CI in consensus-based DKFs involves covariance intersection, with the correction step at sensor node, i, at the end of the final consensus iteration given by information pair

$$q_{k,L}^i = \sum_{j=1}^{Ci}\pi^{i,j}\{(P_k^{j^-})^{-1}\hat{x}_k^{j^-} + H_k^{j^T}(R_k^j)^{-1}y_k^j\} \tag{4.17}$$

$$\Omega_{k,L}^i = \sum_{j=1}^{Ci}\pi^{i,j}\{(P_k^{j^-})^{-1} + H_k^{j^T}(R_k^j)^{-1}H_k^j\} \tag{4.18}$$

where $\sum_{j=1}^{Ci}\pi^{i,j} = 1$, $\hat{x}_k^i = \Omega_{k,L}^i q_{k,L}^i$, and $P_k^i = (\Omega_{k,L}^i)^{-1}$

CM ensures global convergence while CI ensure local consistency. Thus, consensus-based DKF algorithms that implement CM enjoy global convergence, those that implement CI enjoy local consistency, and those that implement both CM and CI enjoy both global convergence and local consistency. To summarise, consensus-based DKFs have good estimation accuracy but high communication requirement.

4.3.2 Diffusions-based DKF Algorithms

Compared to consensus-based DKF algorithms, diffusion-based strategies do not require any knowledge of the network size [59] and neighbouring sensor nodes are involved in a single exchange in between measurements. Thus, the diffusion-based DKF strategy has a low communication requirement. This makes them very appropriate for real-time application in dynamic systems where the dynamics of the system is changing fast (given that measurements can be treated in a timely manner as they are obtained rather than waiting for a consensus on the estimates of neighbouring sensor nodes) and also in systems where the sensor nodes are constrained in energy (as a result of their low communication requirement). Examples of studies where the diffusion strategy of distributed data fusion is applied in WSNs include: [167], [171]–[174].

The diffusion strategy is similar to the consensus strategy with only one consensus iteration between measurement updates. It consists of two steps, an incremental update step followed by a diffusion update step. For a sensor node, i, the incremental update and diffusion update are given by:

Incremental update

$$(P_k^i)^{-1}\psi_k^i = (P_k^{i^-})^{-1}\hat{x}_k^{i^-} + \sum_{j=1}^{Ci} H_k^{j^T}(R_k^j)^{-1}y_k^j \tag{4.19}$$

$$(P_k^i)^{-1} = (P_k^{i^-})^{-1} + \sum_{j=1}^{Ci} H_k^{j^T}(R_k^j)^{-1}H_k^j \tag{4.20}$$

Diffusion Update

$$\hat{x}_k^i = \sum_{j=1}^{Ci} \pi^{i,j} \psi_k^i \tag{4.21}$$

where $\pi^{i,j}$ is an element of the diffusion matrix, with $\pi^{i,j} = 0$ if j $\notin$ C_i and $\sum_{j=1}^{Ci} \pi^{i,j} = 1$

As can be seen in equation 4.21, the diffusion update step of the diffusion strategy is similar to the covariance intersection implemented by the CI variants of consensus-based DKF algorithms. Thus, this gives diffusion-based DKF algorithms the ability to achieve local consistency. However, they do not converge to the CKF value since there is no consensus on measurements. Summarily, diffusion-based DKFs have low communication requirement but their estimation accuracy is poor.

4.3.3 Gossip-based DKF Algorithms

Gossip-based DKF algorithms try to obtain the best from both worlds of consensus-based and diffusion-based DKF algorithms. Unlike the consensus-based algorithms, every sensor node in gossip-based algorithms receives local estimates and performs averaging with only one selected neighbour during each gossip iteration that occur between measurement updates. The selection of a neighbour sensor node with whom averaging is performed during each gossip iteration can be via a randomised protocol [175], a greedy protocol [176], or a sample greedy protocol [177].

Summarily, gossip-based DKF algorithms have a lower communication requirement compared to consensus-based DKF algorithms, but this lower communication requirement is achieved at the cost of slow convergence speed.

Now that we have described the different categories of DKF algorithms, we will in the next section present the selected DKF algorithms that we will implement and evaluate in this study.

4.4 Selected DKF Algorithms

In low-cost WSN applications such as WWPM, criteria such as global convergence and local consistency of the DKF algorithm affect the accuracy of the WWPM system while the communication requirement of the DKF algorithm affects the energy consumption and thus the lifetime of the WWPM system. The effect of global convergence and communication requirement are contradictory to each other. Thus, a comprise is needed in order to achieve acceptable accuracy while prolonging the lifetime of the WWPM system. Given that communications in low-cost

WSNs deplete sensor node's battery faster, the accuracy of a DKF algorithm is emphasized more on its ability to maintain local consistency than its ability to achieve global convergence. Local consistency is of vital importance because inconsistency in fused state estimate and covariance resulting from spurious measurements and the cross-correlations between the local estimates of neighbouring sensor nodes may cause the DKF to diverge [178]. Most DKFs implement the CI technique to handle the cross-correlation between local estimates of neighbouring sensor nodes and thus achieve local consistency. For this reason, we selected only DKF algorithms that maintain local consistency and have low transmission requirement.

Based on the classification and evaluation of DKF algorithms by the study carried out by [59], the following algorithms (shown in Table 4.1) were selected for each distributed data fusion category. They include: Information-Weighted Consensus Filter (ICF) proposed by [64], Sample Greedy Gossip Information-Weighted Consensus Filter (SGG-ICF) proposed by [177], and Event-triggered Diffusion-based Kalman Filter (EDKF) proposed by [171]. All the algorithms selected maintain local consistency as this is very crucial to the reliability of the WWPM system. For the communication requirement, ICF has the highest, followed by SGG-ICF, and lasty EDKF.

Table 4. 1: Comparison of selected DKF algorithms

DKF Algorithm	Underlying fusion strategy	Convergence to CKF	Local consistency	Communication requirement	Fully distributed
ICF [64]	Consensus-based	Yes	Yes	High	No
SGG-ICF [177]	Gossip-based	Yes	Yes	Moderate	No
EDKF [171]	Diffusion-based	No	Yes	Low	Yes

ICF [64] and SGG-ICF [177] both enjoy local consistency (stable estimate for each sensor node in the network) and global convergence (converge to the CKF value in a finite number of iterations) since they implement both CI and CM during the fusion process. However, they are not fully distributed since they need to know the network size during the fusion process [59]. Unlike ICF and SGG-ICF, EDKF [171] is a diffusion-based DKF algorithm since it involves just a single exchange between neighbouring sensor nodes at each time step and the local state and covariance are calculated and then corrected by a convex combination of the estimates of the neighbours [179]. It enjoys local consistency since it implements CI during the fusion process, but does not converge to the optimal CKF value since it does not implement CM. In addition, it is fully distributed, since

information on the network size is not required during the fusion process [59], [171], making it very scalable.

4.4.1 Information-Weighted Consensus Filter Algorithm

The information-weighted consensus filter proposed by Kamal et al. [64] consists of five main steps (initialization, computation of local information pair, information fusion, measurement update and prediction). Every sensor node implementing the algorithm goes through the iterative process shown in Figure 4.1. At time step k, every local sensor node (i) starts by getting inputs such as the predicted state ($\hat{x}_k^{i\,-}$), the predicted state error covariance ($P_k^{i\,-}$), the observation matrix (H_i), the consensus speed factor (ε), the number of consensus iterations (L), the measurement (y_k^i), and the measurement information matrix ($(R_k^i)^{-1}$). This is followed by the computation of the local information pair (u_k^i and U_k^i) by using the inputs received in the initialization step. The next step involves the fusion of local information with those of neighbouring sensor nodes by using a consensus protocol. During each consensus iteration, the sensor node sends its local information pair (u_k^i and U_k^i) to all its neighbours and also receives the local information pair (u_k^j and U_k^j) of all its neighbours. A consensus algorithm is then used to compute the average of each of the element in the information pair during the information fusion step. At the end of L consensus iterations, the fused information pair is obtained ($u_{k,L}^i$ and $U_{k,L}^i$). The fused information pair is then used in the measurement update step to compute the state estimate ($\hat{x}_k^i$) and estimated state error covariance (P_k^i) at time step k. Lastly, the prediction step involves propagating the estimated state and estimated state error covariance in time by at time by computing predicted state ($\hat{x}_{k+1}^{i\,-}$) and the predicted state error covariance ($P_{k+1}^{i\,-}$) at time step k + 1.

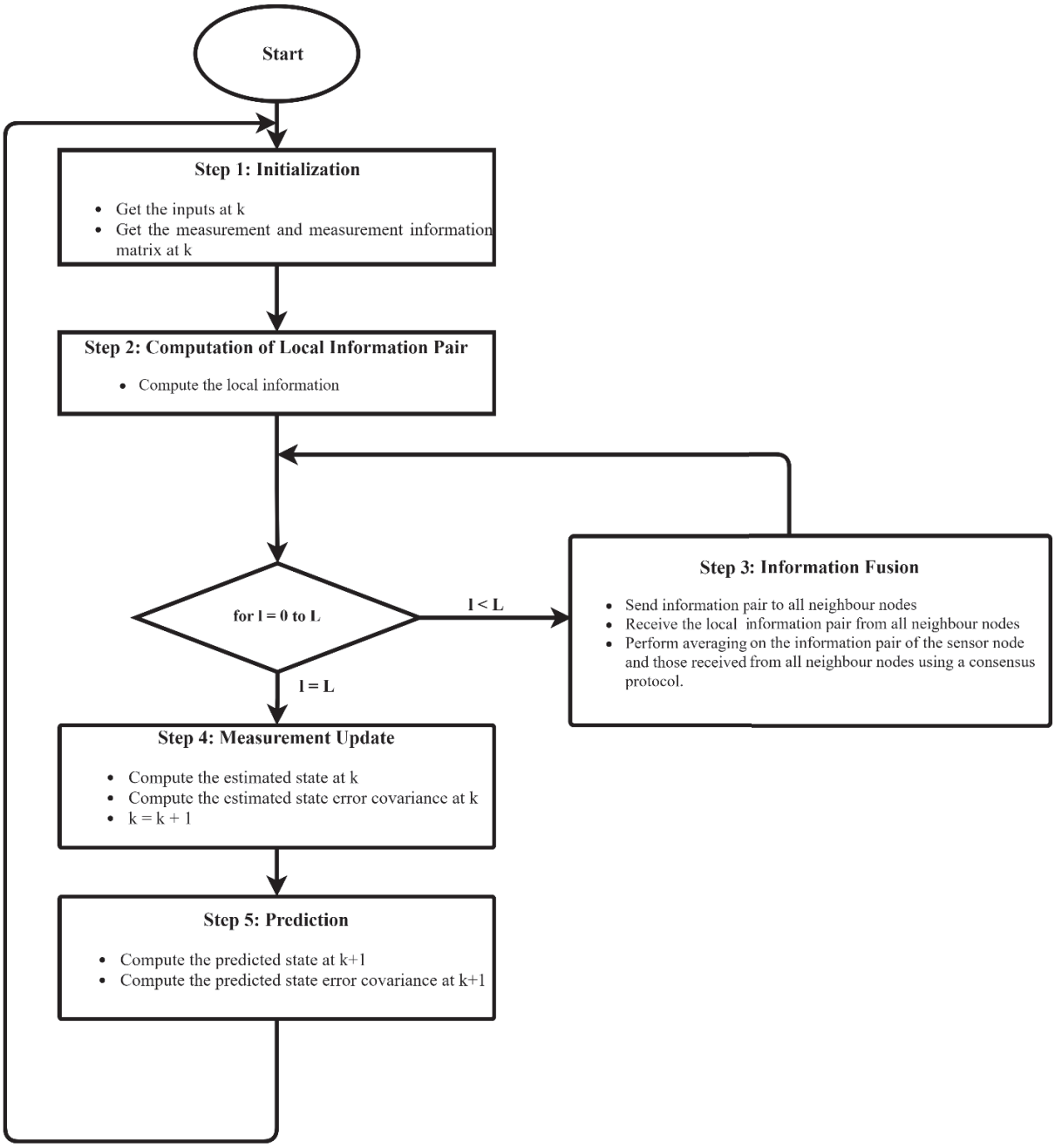

Figure 4. 1: Flowchart of ICF algorithm

4.4.2 Sample Greedy Gossip Information-Weighted Consensus Filter algorithm

The gossip-based DKF algorithm proposed by Shin et al. [177] consists of four main steps (prediction, computation of local information pair, information fusion and measurement update).

Every sensor node implementing the algorithm goes through the iterative process shown in Figure 4.2. At time instant k, every local sensor node (i) computes the predicted state ($\hat{x}_k^{i\,-}$) and predicted state error covariance ($P_k^{i\,-}$) based on the previous state estimate ($\hat{x}_{k-1}^i$) and estimated state error covariance (P_k^i) at k-1, respectively. The next step involves computing the local information pair (u_k^i and U_k^i) which is exchanged with a neighbour node during each gossip iteration. This is then followed by the information fusion step which is repeated iteratively for a total of L gossip iterations. During each gossip iteration, each sensor node first determines a set of active neighbours by generating a probability (p) and comparing it with a sample (q_i) generated from a uniform distribution and stored in each of its neighbours. If the value of the sample stored by the neighbouring sensor node is greater than or equal to the probability generated by the sensor node, then the neighbour sensor is placed into the active set of the sensor node, otherwise the neighbour node is considered as inactive. Once the active neighbour nodes have been determined, the next task performed by the algorithm is to determine for each sensor node the neighbour node with the largest information discrepancy with whom it is going to perform averaging with during the information fusion step. This neighbour node is actually determined via a greedy algorithm that computes the Mahalanobis distance ($d_{i,j}$) between the information pair of the sensor node and those of its active neighbours. The neighbour node, j*, with the largest Mahalanobis distance ($d_{i,j*}$) is selected. The information fusion continues iteratively till after a total of L iterations is reached. After L gossip iterations, the fused information pair information pair ($u_{k,L}^i$ and $U_{k,L}^i$) is available and is used to compute the updated state estimate ($\hat{x}_k^i$) and estimated state error covariance P_k^i at time step k, during measurement update.

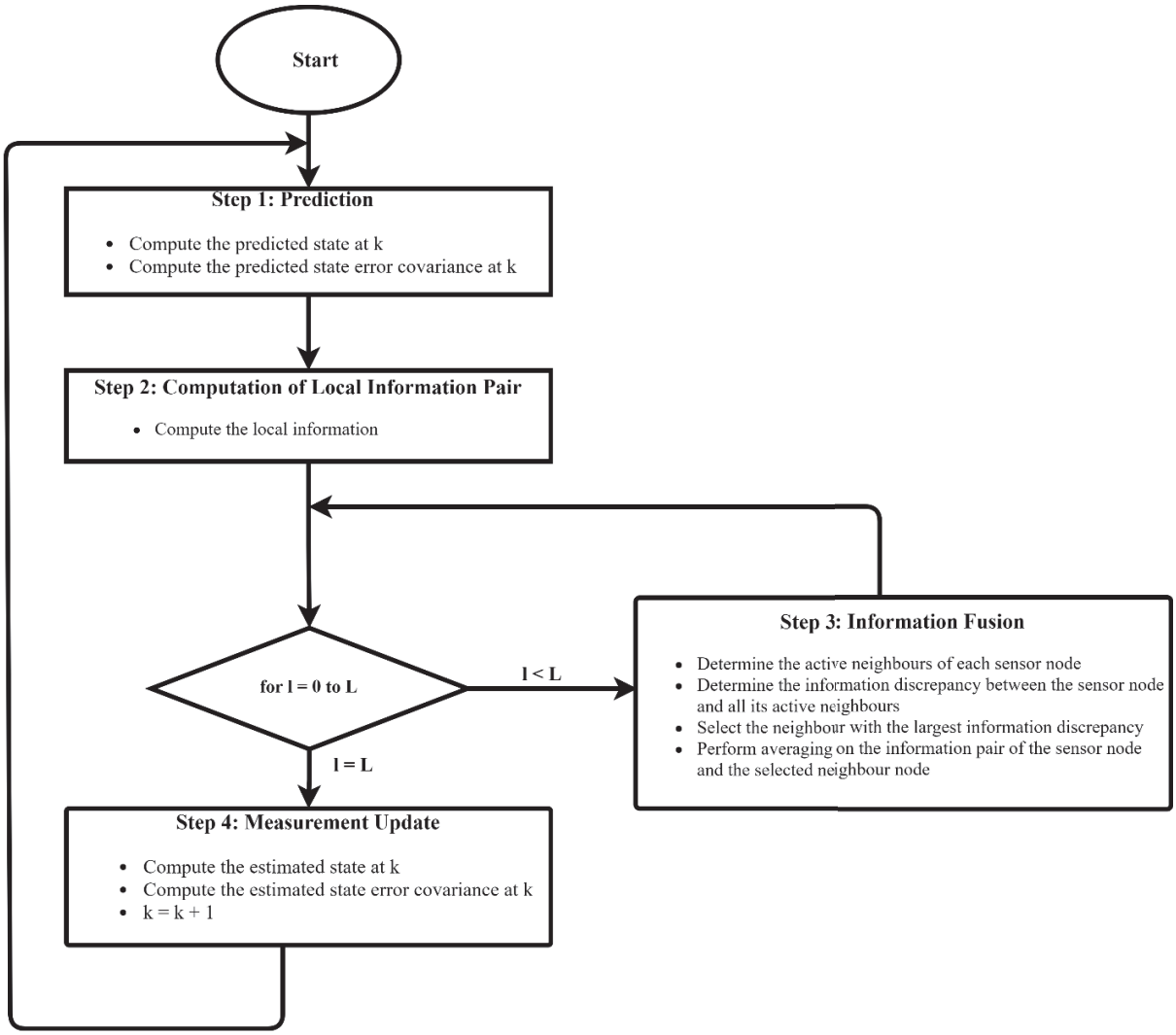

Figure 4. 2: Flowchart of SGG-ICF algorithm

4.4.3 Event-triggered Diffusion-based Kalman Filter Algorithm

The DKF algorithm proposed by Battistelli et al. [171] consists of four main steps (correction, information exchange, information fusion and prediction) and every sensor node implementing the algorithm goes through the iterative process shown in Figure 4.3. Each node starts by updating a local information pair, which consist of the local estimate and the estimation error covariance matrix. This is immediately followed by the exchange of the information pair to the out-neighbours of each node if and only if the transmission flag of the sensor node is set to 1. The transmission flag is set to one when the difference between the current updated local estimate and the last

transmitted local estimate exceeds some threshold, which is determined by some transmission parameters designated in the study as α, β, and δ (which can be varied to achieve a desired behaviour in terms of transmission rate and performance). By this, the algorithm possesses an event-triggered communication capability and information is only shared when the data currently computed by a node's out-neighbours are no longer consistent with the data locally available at the node. The transmission test ensures that in the case of no transmission, the data currently computed by the out-neighbours of a node are close to the data locally available at the node, both in terms of mean and covariance, thus maintaining local consistency. The information exchange step is then followed by an information fusion step where every node computes a fused information pair from its local information pair and those received from in-neighbours at the current time step, k. During the information fusion step, each node computes an approximate local pair for its in-neighbours that did not transmit at time step k (because their transmission flag was not set), from the most recent local information pair last received from them. As the final step in each iteration, the prediction step involves propagating the fused information pair in time by applying the Kalman filter prediction step to compute the local predicted information pair at time k + 1.

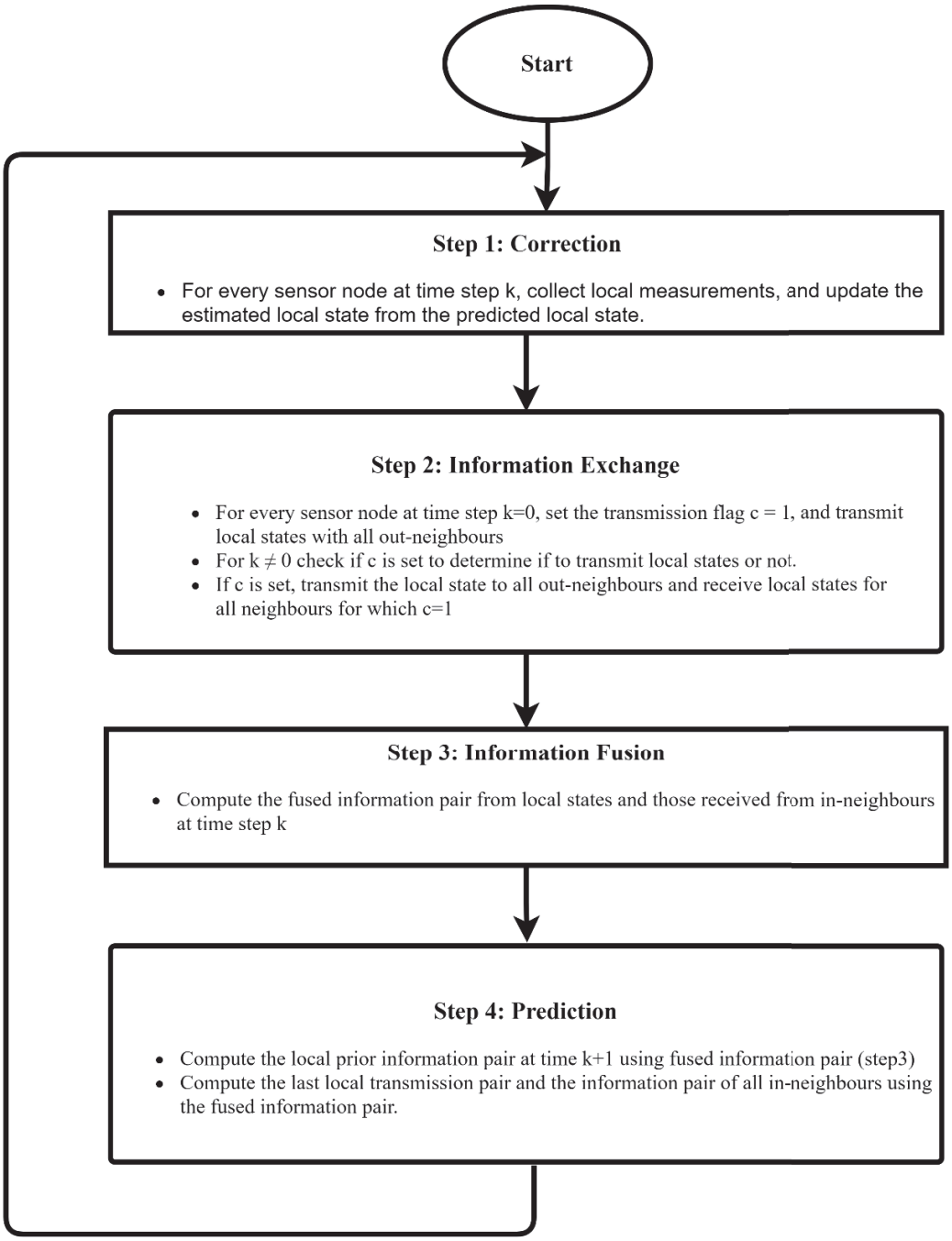

Figure 4. 3: Flowchart of EDKF algorithm

Now that we are done describing the DKF algorithms that will be implemented in this study, we will in the next section present the metrics that will be used to evaluate the performance and power consumption of our proposed WWPM solution.

4.5 Evaluation Metrics

In this section we present the different metrics that can be used to measure the accuracy of leak detection algorithms and energy consumption of WWPM systems. Though information about these parameters is very important especially in the context of WWPM, very few studies provide an evaluation of both the reliability and energy consumption of their solution. While some WWPM studies provide an assessment of the reliability (performance) of their leak detection techniques, very few ever mention the power consumption of their proposed solution. Nevertheless, there exist some studies that have accessed both the accuracy and power consumption of their proposed WWPM solution [40], [41], [51].

4.5.1 Performance metrics

Chan et al. [100], surveyed the different metrics used by WPM studies available in the literature for evaluating the performance of their proposed leak detection algorithms. The performance metrics range from sensitivity, specificity, false alarm rate, precision, accuracy, etc. They presented a confusion matrix (which is mostly used for statistical classification in machine learning), which we adapted to a leak detection system as shown in Table 4.2.

Table 4. 2: Leak detection confusion matrix

		Event	
		Leak	**No leak**
Alarm	**ON**	TP	FP
	OFF	FN	TN

Now, assuming that a ‘leak event’ is a represented by a Boolean ‘1’, a ‘no leak event’ by a Boolean ‘0’, ‘alarm ON’ by a Boolean ‘1’, and ‘alarm OFF’ by a Boolean ‘0’, we generated the leak detection truth table given in Table 4.3.

Table 4. 3: Leak detection truth table

Event	**Alarm**	**Description**
0	0	TN
0	1	FP
1	0	FN
1	1	TP

- TP: represent the cases where there is truly a leak on the WDN and the system triggers an alarm
- TN: represent the cases where there is no leak and the system does not trigger an alarm
- FP: represent the cases of false alarm i.e., cases where the system triggers an alarm but there is actually no leak
- FN: represent the cases where the detection of a leak was missed i.e., a leak actually occurs but the system does not trigger an alarm.

The different metrics that can be used to evaluate the reliability of a leak detection system using the confusion matrix in Table 4.2 include:

1. **Sensitivity**
   It also referred to as the True Positive Rate (TPR) and it represents the ability of the leak detection system to accurately detect all the actual leak events that occur on the WDN and is it expressed as:
   $$Sensitivity = \frac{TP}{TP+FN} \qquad (4.22)$$

2. **Miss Detection Rate (MDR)**
   It is also referred to as False Negative Rate (FNR) and it represents the ability of the leak detection system to fail to trigger an alarm when a leak occurs on the WDN. It is expressed as:
   $$MDR = 1 - Sensitivity = \frac{FN}{TP+FN} \qquad (4.23)$$

3. **Specificity**
   It is also referred to as the True Negative Rate (TNR) and it represents the ability of the alarm of the leak detection system in staying off when there is actually no leak on the WDN. It is expressed as:
   $$Specificity = \frac{TN}{TN+FP} \qquad (4.24)$$

4. **False Alarm Rate (FAR)**
   It is also referred to as False Positive Rate (FPR) and it represents the ability of the leak detection system detect outliers in measurement which do not actually represent the existence of a leak, as a leak on the WDN. It is expressed as:
   $$FAR = 1 - Specificity = \frac{FP}{FP+TN} \qquad (4.25)$$

5. **Positive Predictive Value (PPV)**
   It represents the percentage of leaks events that truly occurred out of the number of leak alarms that were triggered by the leak detection system. It is expressed as:
   $$PPV = \frac{TP}{FP+TP} \qquad (4.26)$$

6. **Negative Predictive Value (NPV)**
   It represents the actual proportion of correctly identified no leakage scenarios in the total number of no leakage scenarios detected by the leak system i.e., how many of the events labelled by the leak detection system as no leakage scenarios were truly no leakage scenarios. It is expressed as:

$$NPV = \frac{TN}{FN+TN} \tag{4.27}$$

7. **Accuracy**
   It is the sum of the correctly identified leakage and no leakage scenarios over the total number of detect events. It is expressed as:

$$Accuracy = \ sensitivity + Specificity = \frac{TP + TN}{TP+FP+ FN+TN} \tag{4.28}$$

One method of detecting the existence of a leak on the WDN is by comparing the value of the measurement at time step k (y_k^i) with the predicted pipe state ($\hat{x}_k^{i\,-}$) obtained from DKF prediction step. If the difference is greater than a certain defined threshold (δ) then a leak alarm will be turned on, else no alarm will be triggered [38], [50], [180].

$$\begin{aligned} y_k^i - \hat{x}_k^{i\,-} &< \delta \quad \text{No leak scenario} \\ y_k^i - \hat{x}_k^{i\,-} &\geq \delta \quad \text{Leak scenario} \end{aligned} \tag{4.29}$$

Another method of determining the presence or absence of a leak on the WDN is by comparing the estimated pipe state at time k ($\hat{x}_k^i$) with a baseline value (x_k^i) of the pipe state when there is no leak on the WDN. If the difference is greater than a certain defined threshold (δ) then a leak alarm will be triggered, else no alarm will be triggered [51], [181].

$$\begin{aligned} \hat{x}_k^i - x_k^i &< \delta \quad \text{No leak scenario} \\ \hat{x}_k^i - x_k^i &\geq \delta \quad \text{Leak scenario} \end{aligned} \tag{4.30}$$

4.5.2 Energy consumption metrics

The energy consumption metric represents the total amount of energy spent by the sensor node while performing sensing, processing and communication operations. It is expressed as:

$$E = \sum_{k} (E_{MCU,k} + E_{TRX,k} + E_{SEN,k}) \tag{4.31}$$

where $E_{MCU,k}, E_{TRX,k}, and\ E_{SEN,k}$ are the energy consumed by the sensor node's processing unit, communication unit, and sensing unit, respectively, at time step k.

$$E_{MCU,k} = I_{MCU} \times V_{MCU} \times t_{proc} \tag{4.32}$$

where I_{MCU} is the current consumption of the MCU, V_{MCU} is the operation voltage of the MCU and t_{proc} is the time used by the MCU for processing.

$$E_{SEN,k} = I_{SEN} \times V_{SEN} \times t_{sen} \tag{4.33}$$

where I_{SEN} is the current consumption of the sensor, V_{SEN} is the operation voltage of the sensor and t_{sen} is the amount of time sensor is active during the sensing process.

$$E_{TRX,k} = \frac{L}{R} V_{TRX}(I_{TX} + I_{RX}) \tag{4.34}$$

where L is the length of the packet transmitted/received, R is the data rate, V_{TRX} is the operational voltage of the transceiver, while I_{TX} and I_{RX} are the current consumption of the transceiver when operating in the transmit and receive modes, respectively.

When the node is powered by a lithium-polymer (Li-Po) battery, the state of charge (SOC) of the battery is estimated and used as the energy consumption metric. There are several techniques for estimating the SOC of a battery. They include coulomb counting method, voltage method, Kalman filter method, impedance spectroscopy, etc [182]. However, we used the voltage method because the battery voltage can be read directly from one of the ESP32 pins without needing extra circuitry. From the SOC versus Voltage discharge curve obtained from the 3.7 V 2000 mAh Li-Po rechargeable battery's datasheet [183], [184], we used polynomial interpolation to derive the analytic relationship between SOC and the voltage of the battery given by equation (4.35).

$$\begin{aligned} SOC = {} & 2808.3808 \times \mathrm{V}^4 - 43560.9157 \times \mathrm{V}^3 + 252848.5888 \times \mathrm{V}^2 \\ & - 650767.4615 \times \mathrm{V} + 626532.5703 \end{aligned} \tag{4.35}$$

where V is the battery's voltage

4.5.3 Selected Evaluation Metrics

Given that the objective of this study was to propose a WWPM system with low-power, real-time, and reliable leak detection, we selected sensitivity, specificity, and accuracy as the metrics for evaluating the performance (reliability) of our proposed WWPM solution. They were selected because they frequently used by most WWPM studies in the literature that evaluated the performance of their solution [35], [40], [51], [181], [185]. This permits us to easily compare the results of our proposed solution with those of existing solutions available in the literature. We evaluated the energy consumption by measuring the current consumption and voltage of the sensor nodes.

4.6 Summary

In this chapter, we reiterated the need for a low-cost, real-time and low-power WWPM solution for detecting leaks on the plastic WDNs of developing countries. Motivated by the need for an algorithm with low complexities (in terms of computational power, storage, and communication) requirement at the sensor node level, we selected KF technology for implementation. We provided reasons why KF was chosen as the signal processing algorithm for processing leak data measured by the low-cost MEMS accelerometers attached to the pipeline. Additionally, we did a survey of different variants of the KF technology that can be implemented in WSNs. We finally ended up using a KF that implements distributed data fusion (DKF) rather than a KF that implements centralized data fusion (CKF) because of the drawbacks of centralized monitoring (implemented by CKF) in WSN which we had highlighted in chapter 2 of this thesis. Guided by the review on DKF algorithms for low-cost WSNs presented by [59], we selected three DKF algorithms for implementation and evaluation in our proposed WWPM solution. One of the selected DKF algorithms is a consensus-based DKF algorithm referred to as Information Consensus Filter (ICF) and proposed by [64]. This algorithm has a good estimation accuracy as it converges to the optimal CKF value and also maintains local consistency of the estimate amongst neighbouring sensor nodes. However, it has a high communication requirement as each node transmits its local information and receives information from all its neighbours at every measurement update and it also requires multiple consensus iteration between measurement updates to achieve a good estimation accuracy. Another DKF algorithm that was selected the event-triggered diffusion-based KF (EDKF) proposed by [171]. This algorithm has a low communication requirement (as each

node transmits its local information and receive information from all their neighbours at every time step and with only one communication iteration involved between measurement updates). This makes it suitable for implementation in WSN applications where there is a strict deadline and the nodes are constrained in energy. However, the estimation accuracy of this algorithm is low. The last algorithm selected is a gossip-based DKF algorithm referred to as sample greedy gossip information consensus filter (SGG-ICF) and proposed by [177]. This algorithm seeks to get the best from both worlds of consensus-based and diffusion-based DKFs by achieving high estimation accuracy while reducing the communication burden. Finally, we ended the chapter by presenting the evaluation metrics that will be used for evaluating the performance of our proposed WWPM solution. We selected sensitivity, specificity, and accuracy as the metrics for evaluating the performance (reliability) of our proposed WWPM solution and we chose to measure the current consumption and voltage of the sensor nodes as the evaluation metric for the power consumption.

4.7 Layout of the Validation procedure

To validate our proposed fully distributed approach for leak detection, it is important to first demonstrate the viability of our DKF-based solution (distributed data fusion) by implementing the selected DKF with the worst theoretical estimation accuracy and comparing the results with those of reference models: LKF-based solution (no data fusion) and CKF-based solution (centralized data fusion). After demonstrating the feasibility of our fully distributed approach, we will then evaluate and compare the performance of the selected DKF algorithms. Thus, in the following two chapters, we will describe the design approach, the deployment and the validation of our distributed approach. We will proceed in two steps. Firstly, we will validate the pertinence of using DKF in WSN with low-cost accelerometers for leak detection. This first step will also be used to validate the global design approach coupling simulation phase and experimental setup. Secondly, we will implement and evaluate the three DKF algorithms selected in Section 4.4 and determine which solution is the best in regards to the selected metrics. Therefore, the next two experimental chapters will be laid out as follows:

1. Demonstrate the feasibility of a DKF-based solution for leak detection in WWPM systems using low-cost MEMS accelerometers by implementing the worst-case scenario, i.e., the selected DKF algorithm with the lowest theoretical accuracy (Chapter 5).
2. Use a combined approach involving simulations on a WSN simulator and physical experiments on a laboratory testbed. This will lead to the construction of a laboratory testbed in Cameroon and the development of an accurate energy model for simulation (Chapter 5).
3. Validate that DKF provides a better leak detection performance when compared to LKF, and it is more scalable and energy-efficient when compared to CKF (Chapter 5).
4. Evaluate the performance and power consumption of the three selected DKF algorithms using the combined approach of simulations and physical experiments once the pertinence of DKF is established. From there, determine which of the selected DKF algorithm is optimal for leak detection in WWPM systems using low-cost MEMS accelerometers (Chapter 6).

Chapter 5

Demonstration of a DKF-based leak detection solution in WWPM Systems using Low-Cost MEMS Accelerometers.

The objective of this chapter is to validate that distributed computing in a WWPM system composed of a network of low-cost MEMS accelerometers for leak detection in plastic WDN is a viable approach. This chapter will be divided in 3 parts: first we will describe the methodological approach composed of a simulation phase and an experimental deployment phase. We will then discuss the implementation of the KF algorithms (LKF, CKF, and DKF) in the simulation and experimental deployment phases. This will be followed by a description of both the simulation setup used for performing simulations and the laboratory setup used for validating the results obtained from simulations. The results of both phases will be presented and discussed in regards to two working hypotheses: comparison with LKF on the one hand and comparison with CKF on the other hand. The first comparison will demonstrate that DKF is viable approach for leak detection and will deliver an efficient simulation model for energy consumption evaluation. The second comparison will show the advantages of this approach compared with conventional centralized approach in terms of energy. We will conclude this chapter by a summary of the obtained results, the necessary evolution of the initial working hypothesis. It should be noted that the results presented in this chapter have already been published in [186], [187].

5.1 Method

5.1.1 Description of Methodological Approach

Our method is a combined approach involving simulations and laboratory experiments. The reason for the combined approach is as follows: we use the simulation results to obtain a first-hand assessment of our proposed solution, then later use the experimental results to validate the simulation results and prove the viability of our proposed solution.

5.1.2 Implementation of KF Algorithms

In this sub-section, we provide the implementation details for the different KF algorithms that were implemented in this first set of experiments. The KF algorithms implemented include: LKF, DKF,

and CKF. LKF and DKF are implemented and evaluated on both the simulation platform and the laboratory testbed while CKF is implemented and evaluated only on the simulation platform. The details of the simulation platform and laboratory testbed are provided in sub-sections 5.1.3 and 5.1.4, respectively. In this first set of experiments, the sensor nodes emulated on the simulation platform and deployed on the laboratory testbed consists of an Adafruit Huzzah32 ESP32 feather as the processing unit [188], Nordic nRF24L01+ as the communication unit [159], and ST Microelectronics LSM9DS1 accelerometer as the sensor unit [161].

5.1.2.1 Implementation of LKF Algorithm

In the implementation of the LKF represented by equations (4.3) through (4.7), we assigned A the value one (**A = 1**). The reason is that steady-state leak detection methods assume that the pipe vibration remains relatively constant until an event such as a leak occurs on the pipeline [189], meaning that the next value of the pipe surface acceleration will be the same as the previous one. Also, we assigned the value of H to one (**H = 1**) since there is a single state of interest (pipe surface acceleration), and it is linearly related to the measurement. The measurement is assumed to be composed of the state value and some noise. Furthermore, with the assumption that there is no control input to the pipeline system, we assigned u_k the value zero ($\boldsymbol{u_k = 0}$). Finally, we derived R from the LSM9DS1 datasheet, and Q was obtained after some experimentation. According to the LSM9DS1 datasheet, the linear acceleration typical zero-g level offset accuracy of the LSM9DS1 accelerometer is ±90 mg, which means the maximum value of R is 0.0081 (**R = 0.0081**). The datasheet of the LSM9DS1 accelerometer states that the sensor when placed in a steady state on a horizontal surface will output 0 g on both the *X*-axis and *Y*-axis, and 1 g on the *Z*-axis. With this in mind, we did some experiments by tuning Q to different values (10, 1, 0.1, 0.01**,** and 0.001) and selected the one that best approximated the acceleration values at zero-g. After the experiments, Q was assigned the value 0.001 (**Q = 0.001**) in the implementation since it best approximated the zero-g acceleration values.

For the simulations, a script written using SenseScript (the language supported by the CupCarbon 4.2 simulation platform) was used to implement the LKF algorithm and finally uploaded to each of the sensor nodes. For the physical implementation, the LKF algorithm was written in C/C++ using the Arduino 1.8.9 Integrated Development Environment (IDE). According to Adafruit

recommendations, the Huzzah32 ESP32 feather should be programmed using either the Arduino IDE or the low-level ESP32 IDF [188]. After compiling the LKF algorithm using the Arduino IDE, the firmware which was uploaded to each of the sensor nodes occupied a memory space of 222 kB.

5.1.2.2 Implementation of DKF Algorithm

We already selected three DKF algorithms in section 4.4 of chapter 4 that we will evaluate in the context of WWPM. However, in this preliminary study, we tested the feasibility of implementing DKF to improve leak detection accuracy in WWPM by first investigating the worst-case scenario (in terms of performance). Amongst the three selected DKF algorithms, we started by implementing the DKF algorithm with the lowest theoretical estimation accuracy. Based on this, EDKF proposed by [171] and described in sub-section 4.4.3 of chapter 4 was implemented as the DKF algorithm in this preliminary study.

The values of the model parameters (A, H, R, and Q) of the LKF algorithm embedded within the EDKF algorithm are the same as those of the LKF implementation earlier mentioned in sub-section 5.1.2.1 above. In addition, the distributed data fusion parameters that determine the information transmission rate of the proposed DKF algorithm, represented in [171] as α, β, and δ, were assigned the values **0.001, 40** and **40**, respectively.

Scripts implementing the EDKF algorithm were written using the SenseScript language and were uploaded to each of the sensor nodes during simulations. For the physical experiments, the nodes were programmed in the same way as in the LKF implementation. Additionally, given that the sensor nodes in the DKF implementation communicate with their neighbours, the RF24 [190] and RF24Network [191] libraries, which provide the MAC and Network layer functions, respectively, were used to control the nRF24L01+ transceiver which was interfaced to the Huzzah32 via SPI. Finally, the firmware uploaded to the nodes after compiling the EDKF algorithm using the Arduino IDE occupied a storage space of 225 kB.

5.1.3 Simulation Setup

5.1.3.1 Selected WSN Simulation Platform

The implementation and deployment of a WSN incurs cost and it is also time consuming. So, it is important to simulate the operation of a specific design before deploying it. In this chapter, we performed simulations to evaluate and provide first-hand information on the performance of the algorithms being studied. Simulations were carried out in Cupcarbon 4.2, which is a Smart City and Internet of Things Wireless Sensor Network (SCI-WSN) simulator that is used to design, visualize, debug and validate distributed algorithms for monitoring, e.g., the collection of environmental data [192]. It offers two simulation environments; one enables the design of scenarios with mobility and the generation of natural events while the other enables the simulation of discrete events in WSNs. It should be noted that CupCarbon simulation is based on the application layer of the nodes, and it is composed of four modules: a microcontroller, radio unit, sensing unit and a battery [193]. It also includes a script called SenScript, which allows the programming and configuration of each sensor node individually. From this script, it is also possible to generate codes for hardware platforms such as Arduino/XBEE [193]. These features make it suitable for simulating distributed algorithms and demonstrating distributed computing in a WSN environment, given that it permits us to write and simulate applications that will be implemented on real sensor nodes. In addition, CupCarbon simulator also provides a feature to monitor the energy consumption and display the detailed energy profile of each sensor node. This can enable us evaluate the power consumption and test the feasibility and realistic implementation of a given distributed solution before its real deployment, as it enables the identification of critical nodes and can give us information on the lifetime of the sensor network. This provides an initial revelation of which algorithm is more energy efficient. This particular feature of CupCarbon simulator is important to our work, as we are interested in evaluating the energy consumption of our proposed distributed solution. Finally, from simulations in CupCarbon, we can also obtain information of the number of messages transmitted which reveals the communication requirement of a given distributed algorithm.

5.1.3.2 Description of Simulation Setup

The simulation setup shown in Figure 5.1a is composed of two sensor nodes (S1 and S2) and natural event generators (A4 and A5). The natural event generator enables the generation of analogue values and its objective is to simulate random or given values from the environment. The simulation setup shown in Figure 5.1b consists of three nodes (S1, S2 and S3). The simulation carried out in the simulation setup depicted in Figure 5.1b was used to show that there is no significant difference in the results obtained from simulating the DKF algorithm in a linear network composed of two nodes and that composed of three nodes. The natural event generators (A4 and A5) were used to simulate acceleration values measured by the LSM9DS1 sensors. One thousand acceleration values generated by the natural events were used in the simulations.

(**a**) (**b**)

Figure 5. 1: Simulation setup in CupCarbon: (a) with two nodes; (b) with three nodes.

5.1.3.3 Simulated Scenarios

We simulated two scenarios using the linear WSN consisting of two sensor nodes (Figure 5.1a). In the first case, we simulated a scenario where the data measured by both sensor nodes (S1 and S2) were erroneous (noisy) while in the second case, only the data measured by sensor nodes S1 were erroneous and the data measured by sensor node S2 were error free. For each case, we implemented both the LKF (without data fusion) and the DKF (with data fusion) and compared the results. The simulation results will be discussed alongside the experimental results in section 5.2 to validate the approach.

5.1.4 Experimental Setup

5.1.4.1 Description of Experimental Setup

We installed a laboratory testbed at the Electrical and Electronic Laboratory of the University of Buea in Cameroon, based on technical and real-field observations of a WDN in Cameroon. The installation is composed of two plastic water storage tanks of capacity 1000 L (one tank for storage placed on a tower of height 9 m and one supply tank placed beneath the tower), a U-shaped 13 m long low-pressure PVC pipe with an external diameter of 32 mm and an internal diameter of 30 mm for the plumbing part. This installation distributes water by using an electrical pump (0.7 HP motor) providing a maximum pump capacity of 40 L/min to fill the upper storage tank. Water is supplied to the WDN by gravity using the upper storage tank, to mimic the WDN of the city of Buea and also to avoid external disturbances caused by operational changes such as the starting/stopping of pumps, etc. Leakage emulation in the pipeline was realized by a valve situated 4 m away from the inlet of the water into the system. The laboratory WWPM system consisted of two sensor nodes, namely S1, placed 1 m before the leak position and S2, placed 1 m after the leak position, as shown in Figure 5.2. In this laboratory setup, the LSM9DS1 accelerometer sensor of each sensor node measured the pipe surface acceleration of the pipe, which was later processed by the ESP32 MCU to detect the presence or absence of leaks in real-time.

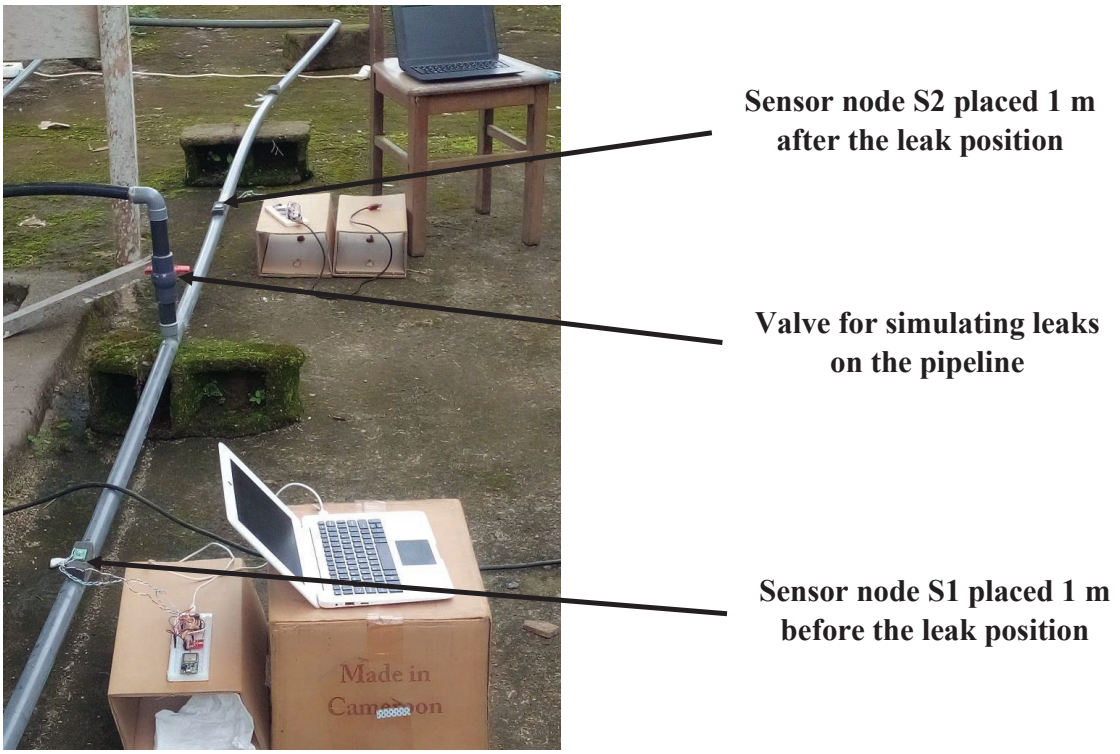

Figure 5. 2: Laboratory testbed setup

5.1.4.2 Validation of the two-node linear WSN implementation

In a chain of sensor nodes forming a linear WSN, which is the case of a WWPM system, the nodes at the beginning and ending of the chain have one directly connected neighbour, while all the intermediate nodes have two directly connected neighbours each. To demonstrate distributed data fusion in a linear WSN, an ideal case is to have a network of at least three nodes, as it represents all the relationships that can be found in a larger linear WSN. However, our laboratory testbed only permits us to perform a two-node physical evaluation. For this reason, we restrained our approach to two nodes for simulation and physical experimentation.

To validate that this two-node approach is still valid, we simulated a DKF algorithm implementation on a two-node linear WSN (results shown Figure 5.3a) and three-node linear WSN (results shown Figure 5.3b). Figure 5.4 depicts a comparison of the root mean square error (RMSE) of node S1 when the simulations were performed in the two-node linear WSN and three-node linear WSN depicted in Figure 5.1. The RMSE of S1 in the two-node linear WSN converges to

0.066 while it converges to 0.064 in the three-node linear WSN. From these results that confirm the strong similarity between Figure 5.3a, b, we can infer that a linear WSN consisting of two sensor nodes can be used to provide a first-hand yet precise evaluation of the performance and validate the feasibility of the distributed approach. Thus, in this preliminary study, we conducted our physical experiments and analysis on a linear WSN comprising of two nodes.

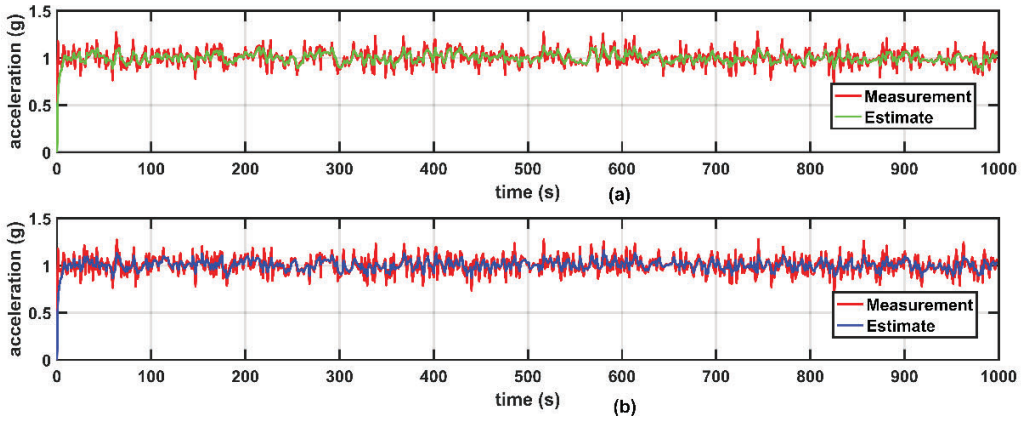

Figure 5. 3: Simulation results for S1: (a) two-node linear WSN; (b) three-node linear WSN.

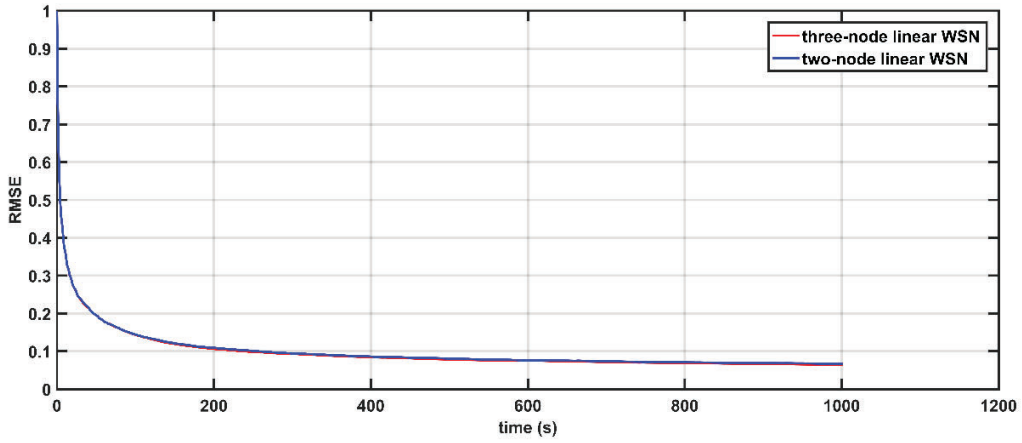

Figure 5. 4: RMSE for node S1 in the case of two-node and three-node linear WSN.

5.1.4.3 Experimental Scenarios

To measure the performance of our leak detection solution based on the EDKF algorithm, we emulated a leak at a single location along the pipeline, as shown in Figure 5.2. In this deployment, the leak location is fixed, but variations in the position of the sensors can be made to evaluate the

effectiveness of our solution at different leak distances. We carried out measurements for two scenarios: an LKF implementation and a DKF implementation. For these experiments, we recorded and plotted traces of data collected from the two sensors, and compared the effectiveness of the approach, i.e., the effect on leak detection when the sensor nodes implemented the EDKF algorithm and in the case where LKF algorithm was implemented on each sensor node.

5.2 Viability of DKF Approach in regards to the LKF Approach

5.2.1 Performance Metrics

We plotted the traces of the estimated acceleration recorded from simulations and lab experiments. The ease of isolating or distinguishing a leakage scenario from a non-leakage scenario on the plotted data was used as the metric to compare the performance of our DKF solution with the reference LKF solution. We did not use the evaluation metrics discussed in section 4.5 of chapter 4 because of the following reasons:

- Most of the reference papers we used when conducting this first set of experiments made used of measurement traces for performance evaluation [13], [38], [41], [58].
- We learned about the performance metrics presented in section 4.5 only after we had already conducted this first set of experiments.

5.2.2 Presentation of Simulation Results and Discussions

In this sub-section, we present and discuss the results of the vibration estimates obtained for the simulation scenarios described in sub-section 5.1.3.3.

Figures 5.5 and 5.6 show the results obtained from simulations when the DKF algorithm was implemented on both nodes (S1 and S2). Figure 5.5 illustrates the results obtained for the scenario where the measurements of both sensors were correlated (noisy to the same extent) while Figure 5.6 illustrates the results obtained for the scenario where the acceleration data measured by both sensor nodes were uncorrelated (measurements from the sensor node (S1) were erroneous and that of sensor node (S2) were error free).

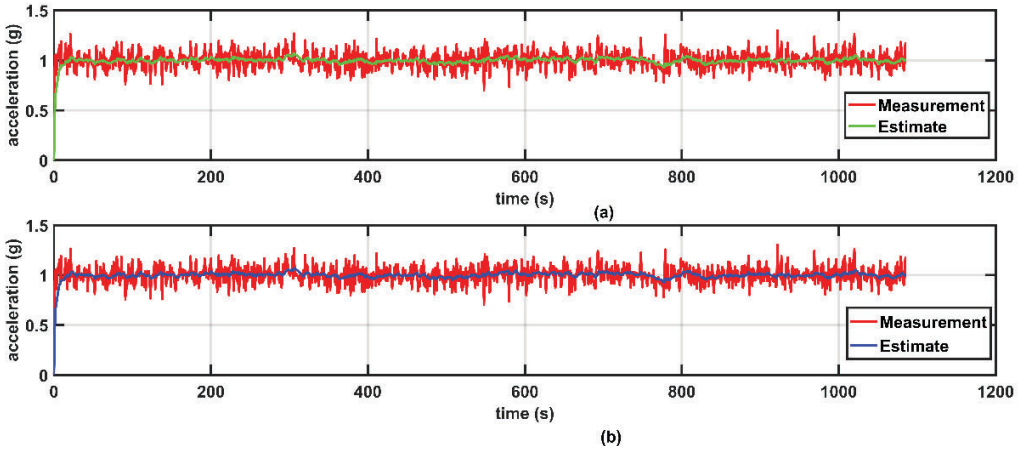

Figure 5. 5: EDKF implementation with noisy measurements from both sensor nodes. (a) Sensor node S1. (b) Sensor node S2.

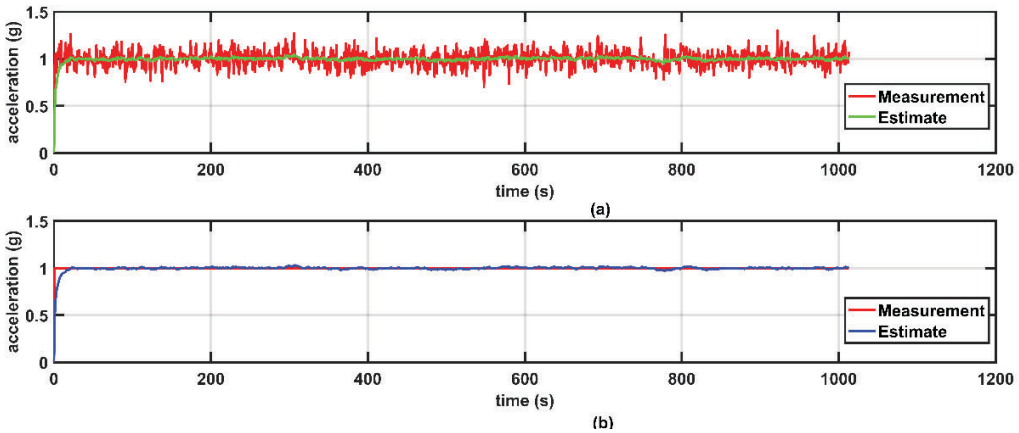

Figure 5. 6: EDKF implementation with noisy measurements from a single sensor node. (a) Sensor node S1. (b) Sensor node S2.

From the results depicted in Figure 5.5, it can be seen that there is no significant improvement in the estimates after filtering and fusion, since the measurements of both nodes are noisy to the same extent in this scenario. This is because when the number of incorrect data sources are greater than the number of correct data sources, the overall performance of the fusion process can be reduced [60], [164]. In Figure 5.6, there is a significant improvement in the estimates of the sensor node (S1) with noisy measurements, since it fuses its local estimate with the local estimate of sensor node S2 (with noise-free measurements). The results of sensor S1 in Figure 5.6 are closer to the true value compared to those in Figure 5.5. However, when the LKF algorithm was implemented, there was no

difference in the results obtained from both scenarios. There was no improvement in the estimates of S1 even when its close neighbour S2 had error-free measurements. This is because there is no fusion of local estimates from neighbouring nodes in the LKF implementation.

The fact that the nodes in LKF implementation perform local processing but do not communicate with each other implies that the LKF implementation consumes lesser energy compared to the DKF implementation. However, its leak detection performance is lower. Additionally, there is no local consistency in the estimates of both sensor nodes in LKF implementation. The implication is that at any given time, one node can be signalling a leak alarm while the other is signalling no leak, thereby deteriorating the reliability of the leak detection system. Therefore, though the LKF implementation provides a low-power solution, to achieve better leak detection accuracy and also maintain local consistency, the nodes must communicate with each other to perform sensor fusion. The simulation results presented reveal the importance of distributed data fusion in improving the accuracy in a fully distributed solution. We validate this assertion via physical measurements in sub-section 5.2.3, where we implement both the LKF and DKF algorithms for leak detection in a laboratory WDN and compare their performance.

5.2.3 Presentation of Experimental Results and Discussions

In this section, we present and discuss the results obtained for the physical experimentation scenarios described in sub-section 5.1.4.3 so as to validate the simulation results presented in sub-section 5.2.2.

Figures 5.7 and 5.8 are results obtained from the laboratory testbed. Figure 5.7 represents the data obtained from sensor nodes S1 and S2 when the EDKF algorithm was implemented on both sensor nodes while Figure 5.8 represents the results obtained when the LKF algorithm was implemented on both sensor nodes.

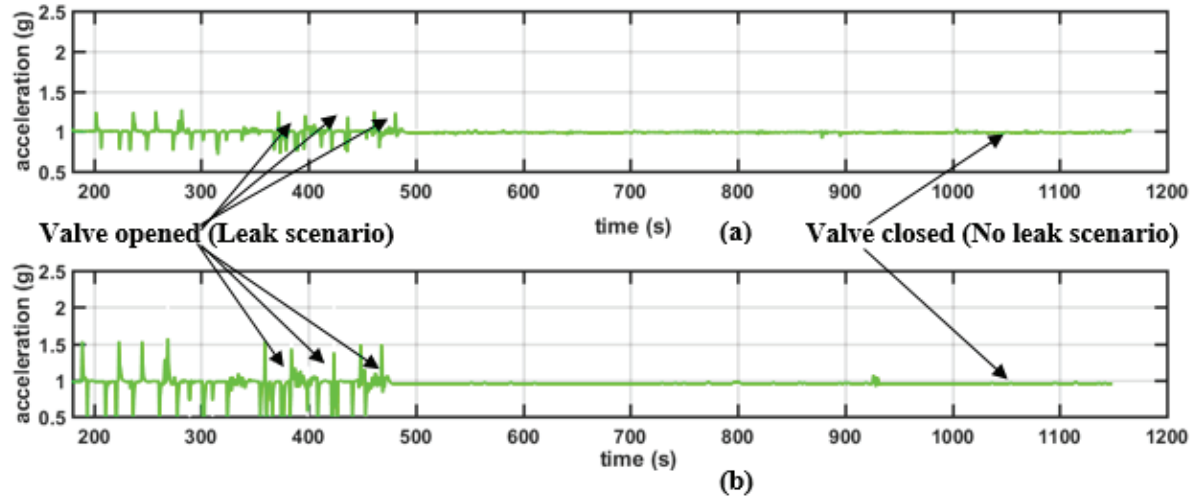

Figure 5. 7: Estimated acceleration from EDKF implementation (a) sensor node S1 (b) sensor node S2.

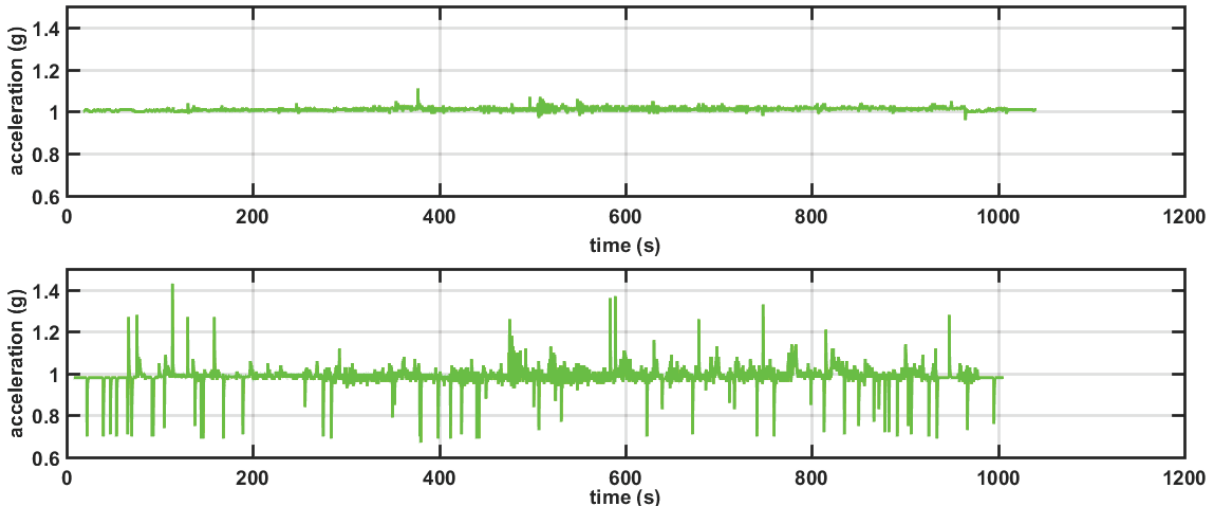

Figure 5. 8: Estimated acceleration from LKF implementation :(a) sensor node S1; (b) sensor node S2.

When there is no leakage, the measured acceleration on the pipe surface is 1.00 g while the estimated acceleration on pipe surface after performing Kalman filtering is 0.99 g. As shown in the results obtained in the field (Figure 5.7), the estimated acceleration of the pipe when there is no leakage is below 1.01 g while an estimated acceleration greater than 1.01 g corresponds to a leakage on the pipe. This is because when there is a leak, there is a fast drop in pressure, leading

to an increase in the flow turbulence which is significantly responsible for the vibrations of the pipe walls, since the source of vibration is dissipated energy caused by turbulence.

Comparing the results in Figure 5.7 (distributed data fusion) with those of Figure 5.8 (no data fusion), reveals that we can easily isolate a leakage scenario from a non-leakage scenario in the DKF implementation. This increases the reliability of detecting leaks and minimizes the rate of false alarms. However, it was difficult to distinguish a leakage scenario from a non-leakage scenario in the LKF implementation. The data depicted in Figure 5.8 have a higher likelihood of producing false alarms since the estimated acceleration computed by the LKF still has a lot of uncertainties. As shown in Figure 5.8, the estimated acceleration is fluctuating rapidly over short time periods. Applying the fixed threshold acceleration of 1.01 g will lead to multiple leak alarms and alarm clears. The reason is that a leak alarm is declared each time the estimated acceleration fluctuates above the threshold value of 1.01 g, and as it fluctuates back below the threshold, the leak alarm clears, resulting in a higher false alarm rate. These results reveal that the DKF implementation provides a higher leak detection accuracy compared to the LKF implementation. The results are also consistent with those obtained in prior literature [97], which experimentally evaluated the performance of DKF and LKF applied to an ultrasound-based positioning application with seven sensor nodes. Additionally, they agree with the results of [45], that showed that the measurements from a single sensor node were insufficient for providing reliable leak detection.

5.2.4 Comparison of Simulation/Experimental Results and Validation of the Approach

In this sub-section, we compare the results obtained from laboratory experiments with the simulation results so as to validate our approach.

An interesting observation from comparing the results obtained from simulations (Figure 5.6) and those obtained from laboratory experiments (Figure 5.7) is the maintenance of consistency in the estimates of sensor nodes S1 and S2. The local consistency is very important in achieving reliable leak detection as it eliminates conflicting leak decisions from neighbouring sensor nodes. This implies that if there is an outlier in the measurements of one sensor node while its neighbour has accurate measurements, the tendency of that false measurement causing a false alarm in the leak

detection system can be minimized by performing distributed data fusion between the neighbouring sensor nodes. Therefore, unlike the solution of Karray et al. [38] where a local fusion centre (cluster head) is required to determine the final leak detection result after the sensor nodes have already performed LKF on the measurements, our solution is fully distributed as there is no need for any fusion centre.

Furthermore, examining the results from simulation and physical deployment, we can draw the conclusion that with the EDKF algorithm implemented, there is a significant of improvement in the reliability of leak detection compared to when the LKF is implemented. This improvement can be attributed to the distributed data fusion capability of the EDKF algorithm. However, one very challenging issue for distributed algorithms in a WSN is their robustness to failed transmissions during the communication process. The results obtained in simulations assumed that no messages were lost during the communications between neighbouring sensor nodes and also did not account for delays due to packet loss. Notwithstanding, the results obtained from simulations were close to what we observed from the physical experimentation. We also observed that in the physical implementation the packet loss rate was very low (<5%). This can be explained by the fact that the distributed data fusion strategy employed by the EDKF algorithm limits the number of communicating nodes for each node to just the directly connected neighbours. Applying this to our context of a linear WSN limits the number of directly connected neighbouring nodes to an upper bound of 2. By properly making use of the transmission time schedule capability of the RF24Network [191] library, the number of collisions is significantly reduced, thus reducing the packet loss rate. In addition to this is the event-triggered capability of the EDKF. The event-triggered nature of the EDKF algorithm reduces the packet transmission rate which also reduces the packet loss rate. From the results presented by the authors in [171], it is clear that the transmission rate of the EDKF algorithm is not uniform over time [171]. The EDKF algorithm has a higher packet transmission rate at the beginning when the estimation error is large. The packet transmission rate reduces when the estimation error is low due to consistency in local estimates and increases again in correspondence with variations in the monitored parameter. Applying this to our WWPM solution implies that the data transmission rate will be very low when there is no

leak on the pipeline and will only increase for a short period of time in the event of a leak appearing on the pipeline.

5.3 Power Consumption Evaluation and Validation of Simulation Model

In this section we present the power profile of the sensor nodes of our DKF-based fully distributed WWPM solution, obtained from simulations and physical measurements. We used the physical measurements to validate the results of our simulated datasheet model. In addition, we present results from physical measurements that show how to reduce the power consumption of the sensor node via duty cycling by using the ULP coprocessor of the ESP32.

5.3.1 Presentation and Discussion of Simulation Results

We used the current consumption data from the datasheets of the different components (ESP32, nRF24L01+, and LSM9DS1) that constitute our sensor node to develop a datasheet model of the sensor node's energy consumption. These data were used in CupCarbon to emulate the power consumption of the sensor node when operating in the different states (sensing, transmitting, computing, receiving, sleeping). Figure 5.9 shows the current profile of the sensor node derived from simulations when the ESP32 is operating at 240 MHz.

To evaluate the energy budget of the node per cycle, we established Table 5.1 from the datasheets of our sensor node's constituent components. As shown in Table 5.1, the sensor node will consume most of its battery energy when it is listening for packets. The reason is that the radio has to be continuously listening for packets since we are dealing with a real-time application where there is a need to minimize the packet loss rate. In sub-section 5.3.3, we present a corresponding table of the node's energy consumption obtained from physical measurements and compare it with those we obtained from simulations.

Table 5. 1: Simulated current consumption of the sensor node at different states.

ESP32 Speed (MHz)	State	Current Consumption (mA)	Duration When Node Is at This State (msec)	Energy Consumption (mJ)
80	CPU idle + radio down	20	50	3.3
	CPU idle + radio listening	31.8	1000	105
	CPU active + radio listening	36.8	900	109
	CPU active + radio transmitting	48.1	50	7.9
240	CPU idle + radio down	40	50	6.6
	CPU idle + radio listening	51.8	1000	171
	CPU active + radio listening	79.8	900	237
	CPU active + radio transmitting	91.1	50	15

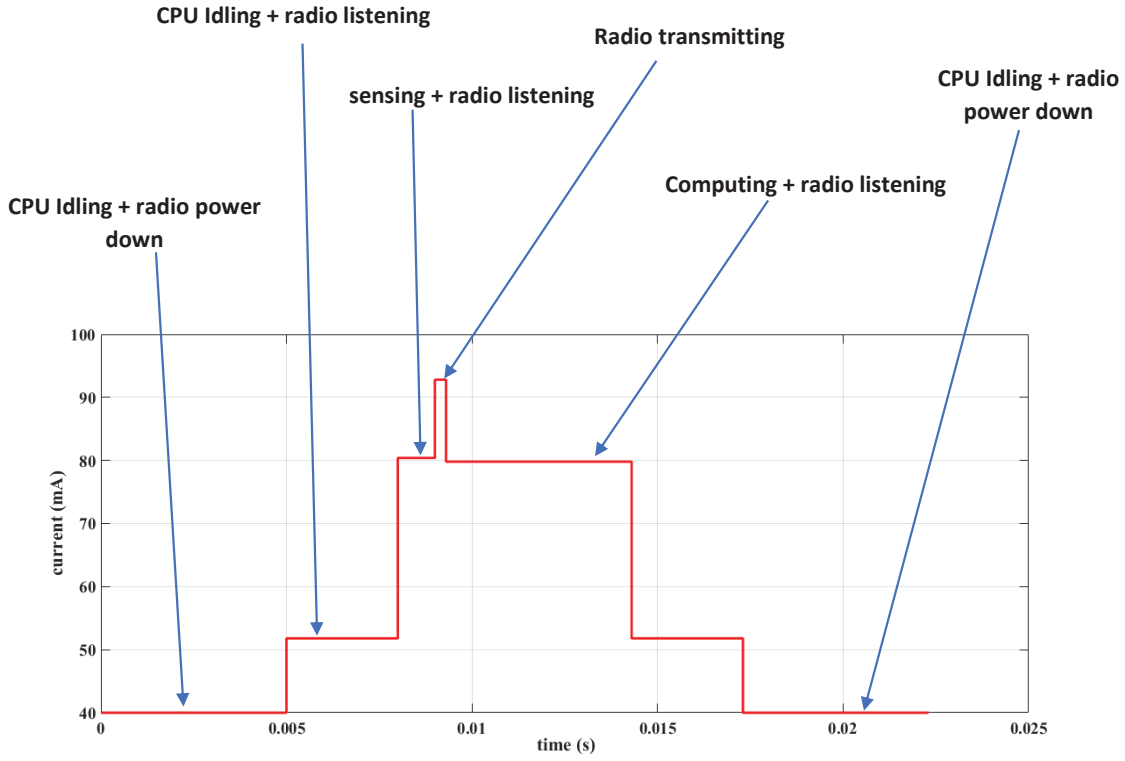

Figure 5. 9: Sensor node's current profile from simulation.

5.3.2 Presentation and Discussion of Experimental Results

We measured the power consumption of the sensor nodes on our laboratory testbed using the custom power measurement device described in section 3.6 of chapter 3. As per the measurements, the average power consumed by the node, depicted by Figure 5.10, is about 100 mW (35 mA at

3.3 V), which is relatively very high for a node that will be powered by a battery. The high-power consumption is explained by the fact that our current implementation uses the ESP32 in modem sleep mode (with current consumption in the range 20 – 31 mA at 80 MHz clock speed) and the radio was always in the listening mode throughout. In modem sleep mode, the ESP32 is actually a power-hungry chip and, as such, is not suitable for battery-powered WSN applications. One way of reducing the power consumption of the sensor node is by reducing the amount of time for which the main core of the ESP32 is active by preferably using the ULP coprocessor for basic control.

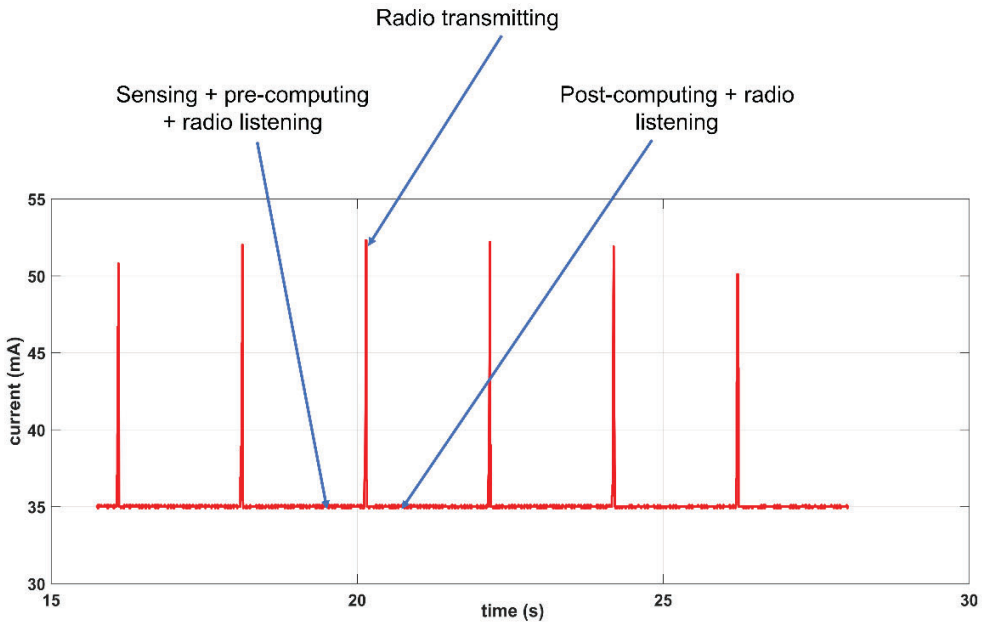

Figure 5. 10: Sensor node's current profile from physical measurements.

To reduce the node's power consumption, we decided to harness the ULP coprocessor of the ESP32 by putting the node frequently into deep sleep mode (with datasheet current consumption in the range 10 µA –150 µA). Thus, the node will only be active for very short periods of time, when it needs to transmit data and perform data fusion. We demonstrated this by using a timer interrupt, which awakened the ESP32 main core when the sleeping period expired. The ESP32 main core was programmed to sleep for 2 s while the ULP coprocessor was functional and it was awakened from sleep using an internal interrupt. Figure 5.11 illustrates the results obtained from putting the ESP32 main core into deep sleep mode while using the ULP coprocessor.

As shown from in the results displayed in Figure 5.11, when the node is in deep sleep mode, the measured current consumption of the node can be as low as 11.8 mA, which corresponds to the current consumed by the nRF24L01+ transceiver in listening mode (11.83 mA), the LSM9DS1 IMU when the accelerometer is operational (600 µA) and the current consumed by the ESP32 in deep sleep mode (10 µA~150 µA). From the results, we can infer that by continuously putting the node into deep sleep mode and only waking it up and keeping it awake for short periods to perform transmission and data fusion with neighbouring nodes, the power consumption of the node can be significantly reduced, thereby increasing the lifespan of the WSN.

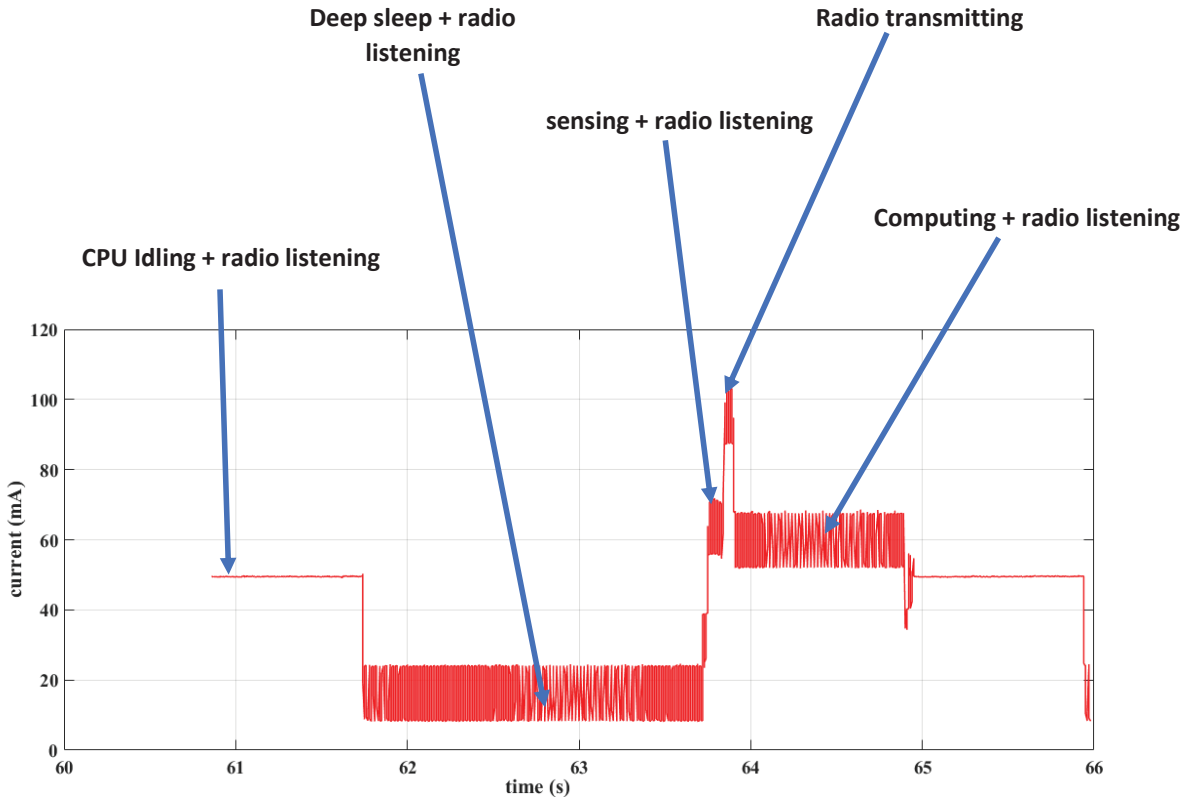

Figure 5. 11: Sensor's node current profile with deep sleep implemented.

5.3.3 Energy Budget Analysis and Validation of Simulation Model

In this sub-section, we perform an energy budget analysis of the sensor node's power consumption obtained from physical measurements and also validate the simulation model of the sensor node's power consumption.

Table 5.2 depicts the energy consumption of the sensor node at different states, obtained from physical measurements, when the ESP32 is operating at 80 MHz and 240 MHz clock speed. It is used for analysis and validation of the energy consumption of the simulation model presented in Table 5.1.

Table 5. 2: Measured current consumption of the sensor node at different states.

ESP32 Speed (MHz)	State	Current Consumption (mA)	Duration When Node Is at This State (msec)	Energy Consumption (mJ)
80	CPU idle + radio down	23.7	N.A.	N.A.
	CPU idle + radio listening	31.8	1000	105
	CPU active + radio listening	35	900	104
	CPU active + radio transmitting	51	50	8.4
240	CPU idle + radio down	39	N.A.	N.A.
	CPU idle + radio listening	50	1000	165
	CPU active + radio listening	69.3	900	206
	CPU active + radio transmitting	102	50	16.8

From the physical measurements taken when the ESP32 of the sensor node was operating at 80 MHz and 240 MHz, we observed that the node operating at 80 MHz is sufficient to run the DKF algorithm and consumes a current which is approximately half of what it consumes when operating at 240 MHz, as shown in Table 5.2. Besides, it is evident that the ESP32 when operating in modem sleep mode at a processing speed of either 80 MHz or 240 MHz, is not energy efficient and not suitable for battery-powered WSN applications. However, as shown in the results in Figure 5.11, proper optimization and harnessing of the ULP coprocessor of the ESP32 while putting the ESP32 core to sleep can significantly reduce the power consumption, thus enabling its usage as a low-end device in battery-powered WSN applications.

Furthermore, as shown in Table 5.2, much of the sensor node's energy is consumed when the radio is listening. The sensor node's radio transceiver (nRF24L01+) has to always be in the listening mode so as to prevent the loss of packets since we are dealing with a real-time application and also because the communications are asynchronous. Even when we harness the ULP coprocessor of the ESP32 to reduce the node's current consumption as shown in Figure 5.11, we still incur much current consumption from the radio transceiver which has to be left in listening mode to prevent the loss of packets. There are two externally-controlled wake-up techniques that can be used to

reduce the energy wasted in idle listening [194]. One way can be to use a radio-controlled wake-up, which incorporates an ultra-low power wake-up receiver to the sensor node's circuitry. By this, we have the possibility of putting the nRF24L01+ transceiver in the power down mode (which consumes 900 nA) while still preventing the loss of packets required for our real-time application. The nRF24L01+ transceiver will only switch from the power-down mode to the listening when there is an interrupt from the wake-up receiver indicating the availability of a packet. Though this will make the sensor node more energy efficient and thus suitable for a battery-powered application, it will nevertheless increase the cost of the sensor node. Another way to reduce the power consumption is to place all the sensor node's components (ESP32 and nRF24L01+) into deep sleep mode and use an external interrupt to awake them from sleep. Given that in our context of fully distributed WWPM, sensor nodes communicate with only their close neighbours to perform fusion, the activity/inactivity feature of the accelerometers (LSM9DS1 or ADXL344) of the sensor nodes can be used to generate an external interrupt (environmentally controlled wake-up) to awaken the sensor nodes from sleep. When there is no leak (no activity) on the pipeline, the ESP32 and nRF24L01+ go into deep sleep mode while the accelerometer (LSM9DS1 or ADXL344) is active. Once a leak (activity) occurs, it is detected by the accelerometer, which then sends an external wake-up to both the ESP32 and nRF24L01+. This way we can achieve low-power consumption while also preventing the loss of packets, because neighbouring sensor nodes will wake up at almost the same time given that they are all within the vicinity of the event. Additionally, since the LSM9DS1 consumes more energy than the ADXL344 and it is also more accurate, the ADXL344 can be used for activity detection to awake the LSM9DS1, ESP32, and nRF24L01+ from deep sleep mode, after which the LSM9DS1 can then collect accurate readings which are used by the leak detection algorithm to confirm the presence of a leak on the pipeline. Thus, by using duty cycling and hierarchical sensing, the WWPM solution can achieve both real-time leak detection and low-power consumption.

Finally, when we compared the power consumption of the sensor node obtained from the datasheet model with those obtained from physical measurements, we realize that the power consumption of the node derived from the model was close to that obtained from physical experiments. For instance, the current consumption of the sensor node when transmitting is 91.1 mA derived from

the model and 102 mA obtained from physical measurements. From these results, we realize that the power consumption of the sensor derived from simulation can provide us with a first-hand, yet precise approximation of the power consumption of the sensor nodes when embedded with a given distributed algorithm we intend to study before physical implementation. This can be used to quickly evaluate the power consumption of a distributed algorithm and its influence on the lifespan of the WSN before physical implementation. Hence, it can be used to predict which distributed algorithm is more energy efficient. This model will be greatly utilized in our future experiments where we will evaluate all the selected DKF algorithms for their power consumption and their performance in reliably detecting leaks.

From the results presented in sections 5.3 and 5.4, we have shown practically that DKF can be used for leak detection in plastic WDN using WWPM systems composed of low-cost MEMS accelerometer sensors. We have also validated the combined approach (simulations and laboratory experiments) and developed a simulation model for the power consumption. In the following section, we will evaluate the scalability of DKF in a large-scale deployment.

5.4 Performance Comparison between DKF and CKF

5.4.1 Aim

Now that we have demonstrated via simulations and physical experiments that the distributed data fusion capability of a DKF can be of interest in improving the reliability of leak detection using low-cost MEMS accelerometer sensors, and also established a model for the power consumption and validated it via physical experiments, we intend to extend our solution to a large-scale WSN. The goal is to measure interesting elements such as the lifetime of the network and also carry out real large-scale analyses that compare the scalability of CKF and DKF.

5.4.2 Method

In this section, we compare the power consumption and communication requirement of the implemented DKF algorithm (that implements distributed data fusion) and the benchmark CKF algorithm (that implements centralized data fusion) in a large-scale WSN. The WSN consists of 10 sensor nodes (S1–S10) connected in a linear topology and it was simulated in CupCarbon as shown in Figure 5.12. We used the datasheet energy model developed in sub-section 5.3.1 and validated in sub-section 5.3.3 to compare the power consumption of the implemented DKF

algorithm and the benchmark CKF algorithm. For the power consumption metric, we displayed the results of the state of the battery for each sensor node for both the DKF and CKF implementations. For the communication requirement, we provided results for the number of messages transmitted for both the DKF and CKF algorithms.

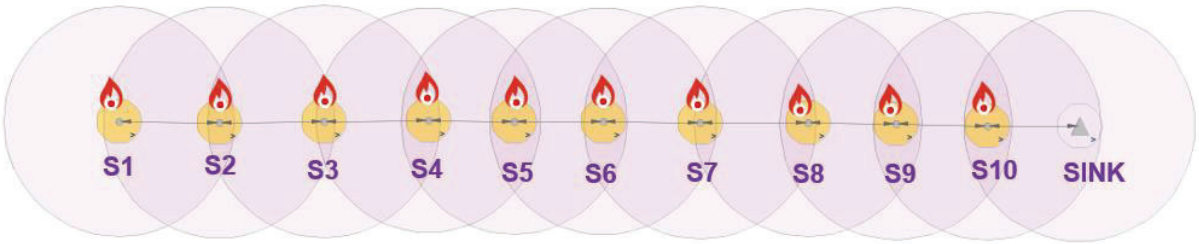

Figure 5. 12: Simulation of a linear WSN consisting of ten sensor nodes and one sink node.

5.4.3 Results and Discussions

In the distributed implementation, the sink node is not involved in the fusion, whereas in the centralized implementation, the sink node is involved in the fusion of the data obtained from all the sensor nodes. This requires numerous multi-hop communications and the sensor node (S10) directly connected to the sink node is involved in relaying the data from all the other sensor nodes to the sink. This makes S10 a critical node in the centralized implementation since it has a higher probability of developing an energy hole, which will affect the lifespan of the WSN. Figures 5.13 and 5.14 display the simulation results of the sensor nodes' energy profiles for distributed and centralized solutions, respectively, for a simulation time of 1 day (86,400 s).

Comparing the energy consumption of the DKF algorithm (Figure 5.13) with that of CKF (Figure 5.14), we realize that the energy consumption of the implemented DKF is less than that of the benchmark CKF. From Figures 5.13 and 5.14, it can be seen that at time t = 80,000 s, none of the sensor nodes in the DKF implementation have a battery state below 1.8×10^4 J, whereas all the sensor nodes except S1 in CKF implementation had crossed that state at simulation time t = 80,000 s. S1 is an exception because it is involved in less multi-hop communications compared to the other nodes. From the results obtained, we observed that for a day period (86,400 s) sensor node S10 had exhausted 47.8% and 3.6% of its battery energy in the CKF and DKF implementations, respectively. We also observed that the nodes' energy consumptions in the DKF are almost

uniform. This can lead to an extension of the lifespan of the WSN since the likelihood of an energy hole developing at a sensor node when the majority of other nodes in the WSN still have sufficient energy is low. However, in the CKF implementation, the nodes' energy consumptions are not uniform. For example, there is a greater likelihood of an energy hole occurring at S10, thereby shortening the lifespan of the WSN.

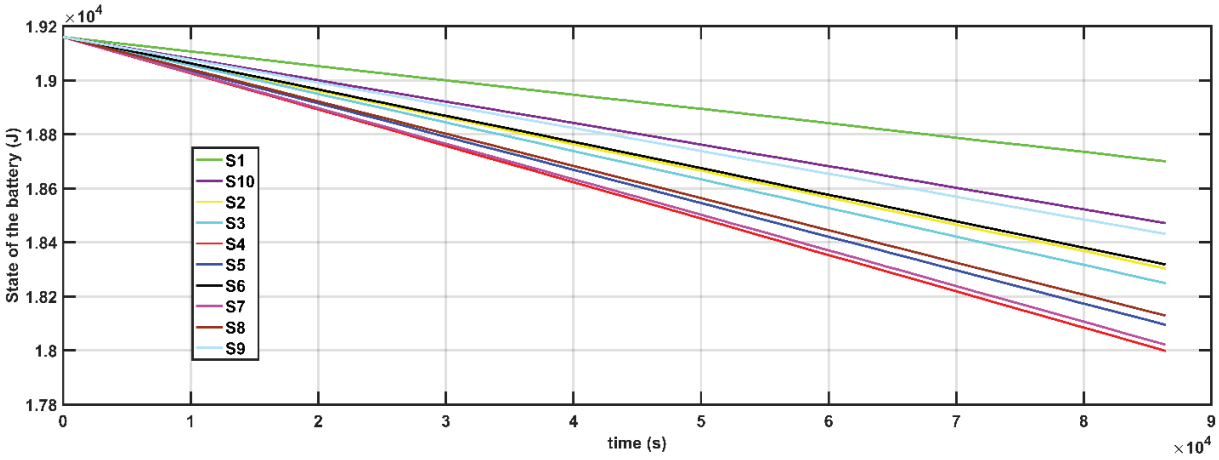

Figure 5. 13: State of the battery of sensor nodes for the distributed implementation.

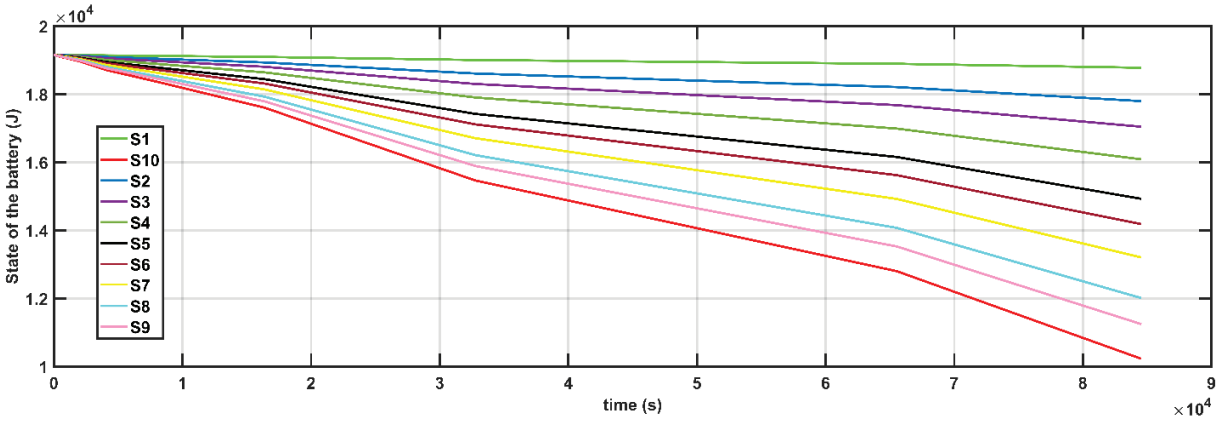

Figure 5. 14: State of the battery of sensor nodes for the centralised implementation.

A comparison of the communication requirement of the DKF and CKF implementation obtained from CupCarbon revealed that 198,380 and 3,265,479 messages were transmitted for the case of DKF and CKF implementations, respectively. From this simulation results, we observed that the communication requirement of CKF is about 16 times higher than that of the implemented DKF.

Though the CKF algorithm is Bayesian optimal, it has a very poor performance when it comes to energy consumption and bandwidth utilization. Other drawbacks are its lack of scalability, high degree of latency, and lack of robustness. This makes it an infeasible solution for large-scale WSNs.

5.5 Summary

In this chapter, we demonstrated and evaluated a fully distributed solution for leak detection in WSN-based Water Pipeline Monitoring (WWPM). With most of the state-of-the-art studies of distributed computing in WSN being theoretical and validated by simulations, we in this chapter demonstrated the feasibility of a reliable DKF-based solution for leak detection in WWPM systems composed of a network of low-cost MEMS accelerometers by using a combined approach that involved simulations on CupCarbon 4.2 simulator and physical experiments on a WDN laboratory testbed. We showed via simulations and laboratory experiments how DKF improved the accuracy and extended the lifespan of a WWPM system by implementing the DKF algorithm proposed by Battistelli et al. [171]. In our proposed solution, all the processing required for leak detection is performed at the sensor nodes, without needing a centralized base station for the processing of leak signals. The implemented DKF algorithm was used for processing the vibration signals read by low-cost MEMS accelerometers attached to the pipe surface in order to detect the occurrence of leaks on the pipeline. Initially, simulations and physical experiments were first performed on a two-node linear WSN, where the leak detection performance of the implemented DKF were compared with the reference LKF implementation. This was later followed by simulations and laboratory experiments that evaluated the power consumption of our proposed DKF-based solution and validated the simulation model for energy consumption evaluation. Finally, simulations were conducted on a larger WSN consisting of ten linearly connected sensor nodes, where the power consumption of the implemented DKF was compared with that of the benchmark CKF algorithm. Our results showed the feasibility of applying a DKF-based fully distributed solution for reliable leak detection in WWPM systems composed of a network of low-cost MEMS accelerometers. In addition, the results established the importance of distributed data fusion in improving the reliability and power consumption of the leak detection system. From the physical implementation, the results showed that distributed data fusion implemented by the DKF increases the reliability

of leak detection when compared to the LKF while simulations on the ten-node linear WSN showed that the implemented DKF significantly preserved the WWPM lifetime when compared to the CKF.

Now that we have demonstrated that DKF works and that it is an optimal solution in terms of reliability and energy consumption trade-offs, we will need more sophisticated performance metrics than just mere visualization of estimated acceleration traces used in this chapter to provide a better comparison of the performance of the three selected DKF algorithms. Thus, we will in the next chapter use the performance metrics (sensitivity, specificity, and accuracy) developed in section 4.5 of chapter 4 to compare the leak detection performance of the three selected DKF algorithms discussed in section 4.4 of chapter 4. Besides, the power consumption results in this chapter showed that in our proposed fully distributed solution, much energy is wasted on idle listening to maintain real-time and reliable leak detection. To curb the power consumption while maintaining real-time and reliable leak detection, we will in the next chapter implement duty cycling and hierarchical sensing at the sensor node level. Lastly, we realized that leak characterization is very important in order to determine the maximum sensor spacing and the minimum size of the leak that can be detected. Thus, in the next chapter, we will first perform leak characterization on a slightly modified WDN laboratory testbed before we perform experiments to evaluate the performance of the three DKF algorithms.

Chapter 6

Evaluation of the Leak Detection Performance and Power Consumption of the Three Selected DKF Algorithms.

The first objective of this chapter is to implement the three selected DKF algorithms. The second is to compare their leak detection performance and power consumption and determine which of the three DKF algorithms is optimal for leak detection in WWPM systems composed of a network of low-cost MEMS accelerometer sensors. This is important because the three DKF algorithms implement different distributed data fusion strategies, which affect both the reliability and power consumption of the fully distributed WWPM system. Therefore, the challenge is to determine which DKF solution provides a better leak detection performance and consumes less energy. The third objective is to demonstrate how the power consumption of the proposed fully distributed solution can be reduced via duty cycling and hierarchical sensing.

This chapter starts by presenting the methodological approach (similar to the combined method used in Chapter 5 but with slight modifications to the experimental setup) and the implementation of the selected DKF algorithms. This is followed by a presentation of the results of the leak characterization experiments performed on the laboratory testbed, where the pipe surface acceleration is measured at different distances from the leak position and for different leak sizes. Next, we present the results of simulations and laboratory experiments, evaluate the leak detection performance and power consumption of each DKF implementation, and compare the results obtained from simulations with those derived from experimentation. From the performance and power consumption results, we determine the optimal DKF algorithm for leak detection based on a performance and energy consumption trade-off. Finally, we end this chapter by presenting the results of power consumption reduction achieved for experiments conducted on the laboratory testbed, where we implemented hierarchical sensing and duty cycling at sensor node level.

6.1 Method

We used the combined approach described in Section 5.1. The simulation setup is the same as in sub-section 5.1.3, but we did some slight modifications on the laboratory testbed cf. sub-section 5.1.4.

6.1.1 Implementation of Selected DKF Algorithms

Similar to the experiments in chapter 5, the simulation scripts were written using the SenseScript language in CupCarbon while for the laboratory experiments, the firmware uploaded to the nodes were written in C/C++ and compiled using the Arduino IDE version 1.8.9. The KF parameters A, H, Q, and R were assigned the values 1, 1, 0.001, and 0.081, respectively. Table 6.1 provides a summary of the values of the parameters assigned to the various DKF algorithms.

Table 6. 1: Values assigned to DKF parameters

Parameter	Value	Concerned algorithms
State transition matrix (A)	1	All
Measurement matrix (H)	1	All
Process noise covariance (Q)	0.001	All
Measurement noise covariance (R)	0.0081	All
Network size (N)	2	ICF and SGG-ICF
Number of consensus or gossip iterations (L)	5	ICF and SGG-ICF
Consensus speed factor (ϵ)	0.65	ICF
Sensor activation probability (p)	0.5	SGG-ICF
Information transmission rate (α, β, and δ)	0.001, 40, 40, respectively	EDKF

For the ICF algorithm, the value of the network size, N was set to 2 (since we are working on a two-node linear WSN), and the number of consensus iterations (L) was set to the value 5 for both the simulation and the laboratory experiments. The number of consensus iterations was chosen to be 5 based on the results of He et al. [59] which revealed that the ICF algorithm converges asymptotically to the CKF value and also achieves local consistency with the number of communication iterations, L, having the value of 5. The consensus speed factor (ϵ) was assigned the value 0.65 in both the simulation and laboratory implementations. For the laboratory experiments, the firmware uploaded to the nodes after compiling the ICF algorithm using the Arduino IDE occupied a storage space of 232 kB.

For the SGG-ICF algorithm, the value of the network size N was also set to 2, and the number of gossip iterations (L) was set to the value 5 for both the simulations and the physical experiments. The sensor activation probability (p) was assigned the value 0.5. The Arduino sketch implementing the SGG-ICF algorithm that was uploaded to the sensor nodes also occupied a storage space of 232 kB.

The implementation of the EDKF algorithm is the same as described in sub-section 5.1.2 of chapter 5 for both the simulations and laboratory experiments.

Now that we have described the implementation of the DKF algorithms, we will in the next section provide a detailed description of the simulation and experimental setups and the simulations and experiments conducted on the simulation platform and the laboratory testbed, respectively.

6.1.2 Simulation Setup

6.1.2.1 Description of Simulation Setup

The simulation setup shown in Figure 6.1 is composed of two sensor nodes (S1 and S2) and natural event generators (A4 and A5). The natural event generators emulated the physical accelerometer sensors and were loaded with acceleration data collected from the laboratory testbed. The sensor nodes (S1 and S2) were loaded with scripts that implement the selected DKF algorithms during the simulations.

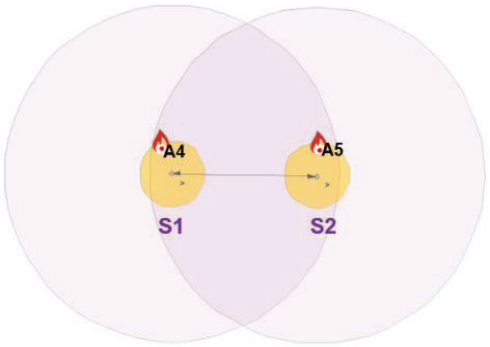

Figure 6. 1: Simulation setup in CupCarbon

6.1.2.2 Simulations

We conducted several simulations on the setup depicted in Figure 6.1, to assess the performance and power consumption of each of the selected DKF algorithm. Firstly, we ran simulations to evaluate the estimation accuracy of each DKF algorithm by computing the root mean squared error (RMSE). We then performed simulations to evaluate the leak detection performance of each DKF algorithm by loading the sensor nodes with datasets that contained a known number of leak and the no leak events. We finally ended by conducting simulations that ran for a period of 86400 s (1 day) to monitor the power consumption of the nodes when executing each of the selected DKF algorithm. The results of these simulations are presented and discussed in Sections 6.3 and 6.4.

6.1.3 Experimental Setup

6.1.3.1 Description of the Experimental Setup

The laboratory testbed used for this second set of experiments was similar to that described in sub-section 5.1.4. However, high-pressure PVC pipes having an outer diameter of 25 mm were used for the laboratory WDN setup instead of low-pressure PVC pipes having a diameter of 32 mm that were used in the first set of experiments presented in chapter 5. The reason for the change was because pressure pipes, though expensive are commonly used in the WDN of Cameroon due to their low susceptibility to external damages and high resistance to burst. Our aim in this second set of experiments was to use pipe materials commonly used in real life WDNs.

The laboratory WDN setup was made up of two 25 mm diameter pressure pipes, with each having a length of 6 m, joined together to produce an L-shaped structure. Two valves were installed on the pipeline, one at the end of the pipeline which is intended to act as the service valve to emulate water consumption at the client's premise and one placed 8 m away from the inlet of the water into the distribution pipe that acted as the leak valve to emulate leaks in the WDN. Based on our assumption that there are no vibrations generated by the opening/closing of taps at the client's premise, the service valve remained closed throughout these experiments. The only vibrations in the pipeline system were assumed to be vibrations caused by leaks emulated by opening the leak valve. Figure 6.2 below illustrates the laboratory testbed setup.

(a) **(b)**

(c) **(d)**

Figure 6. 2: Laboratory testbed setup (a) Distribution tank placed at a height of 9 m (b) Supply tank found beneath the tower (c) distribution pipeline (d) leak valve

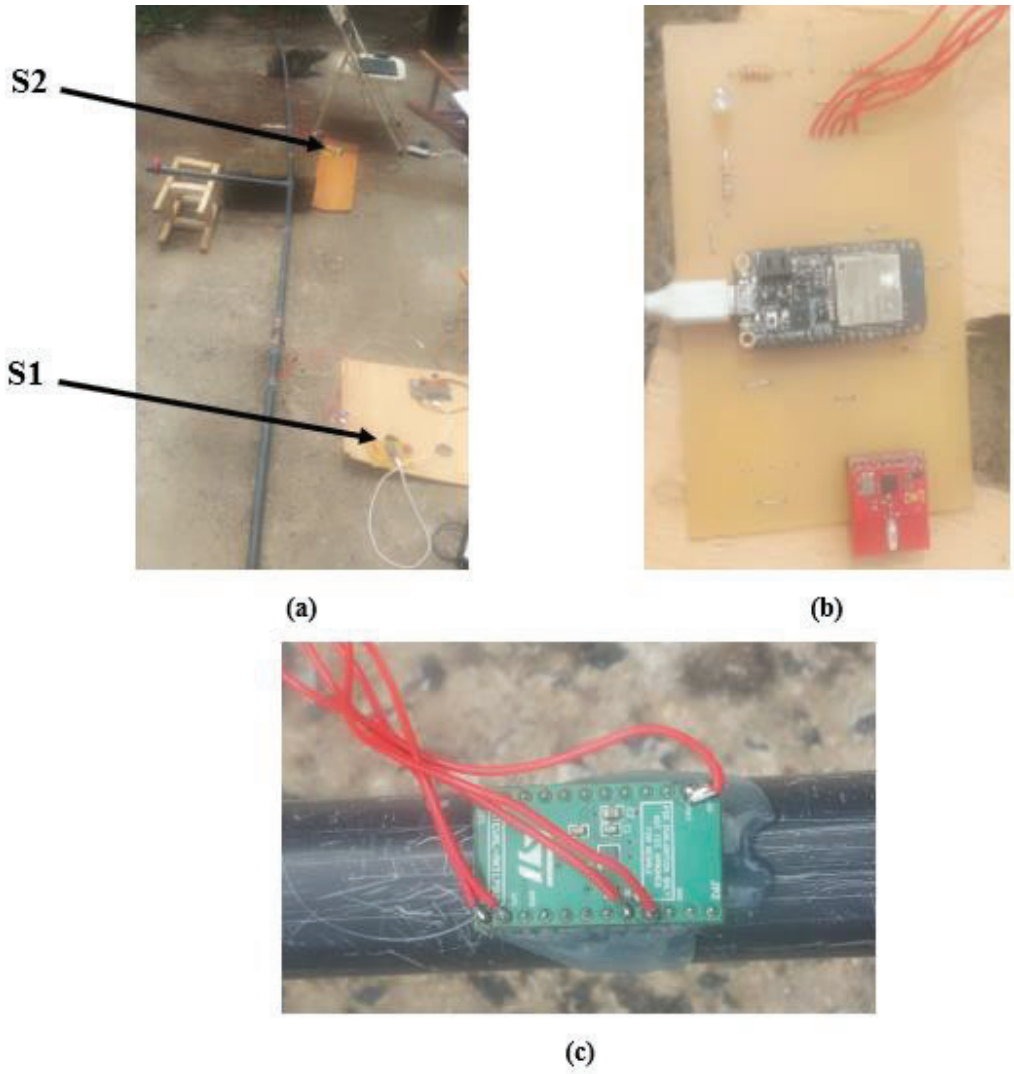

Figure 6. 3: Two-node linear WWPM system (a) Position of sensor node's S1 and S2 (b) Sensor node on PCB (c) Mechanical coupling of the accelerometer to the pipe surface

The laboratory WWPM system was composed of two sensor nodes S1 and S2, which were placed before and after the leak valve, respectively, in the direction of water flow (left to right) as shown in Figure 6.3. The composition and interfacing of the sensor node components is the same as those used in the first set of laboratory experiments, with detailed description found in sub-section 5.1.2. The main core of the ESP32 MCU on each sensor node was configured to operate only at a speed of 80 MHz. The accelerometers of both sensor nodes were glued on the pipe surface using hot glue and wires were used to connect them to the ESP32 MCU. Figure 6.3c shows the mechanical

coupling of the accelerometer and the pipe surface. Hot glue was used because it provided good mechanical coupling between the accelerometer and the pipe. In addition, it made it easier to remove the sensor from the pipe surface without destroying the sensor.

6.1.3.2 Laboratory Experiments

Firstly, we conducted experiments to characterize the leak by varying the distance of the sensor from the leak position and also varying the size of the leak. The results and discussion of the leak characterization experiments are found in section 6.2. Secondly, to measure the leak detection performance of the selected DKF algorithms, we emulated a leak on the pipeline by opening the leak valve at specific times and then recorded the number of times each DKF implementation triggered an alarm when the valve was opened (true positive), failed to trigger an alarm when the valve was opened (false negative), triggered an alarm when the valve was closed (false positive), and when it did not trigger an alarm when the valve was closed (true negative). The results and discussions of the conducted leak detection performance experiments are found in section 6.3. Finally, we conducted experiments to measure the power consumption of each of the selected DKF algorithm. We used the USB power meter described in section 3.6 of chapter 3 to monitor the node's power profile for each of the DKF algorithm for a period of 1 hour. We later supplied each node with a 3.7 V, 2000 mAh Li-Po battery and monitored the state of the battery discharge for a period of 1 day. The results and discussions of the power consumption of each of the DKF algorithms are found in section 6.4. Finally, we conducted experiments where we evaluated power consumption reduction by implementing hierarchical sensing and duty cycling at sensor node level. The results and discussions of the power consumption reduction are found in section 6.6.

6.2 Leak Characterization

In this section, we study the influence of the distance from the leak position and the size of the leak on the vibration data collected from the pipe surface using the LSM9DS1 accelerometer. The objective is to determine the maximum distance from which the accelerometer sensor should be placed from the leak position on the pipeline to effectively detect the presence or absence of a leak and the size of the leak that can be detected by the accelerometer. To study how the distance from the leak position affects leak detection, we varied the distance of the sensor from the leak position and collected vibration data at distances of 0.25 m, 0.5 m, 1 m, and 2 m from the leak position. For

each distance from the leak position, we varied the leak size by tuning the size of the leak valve, measured the flowrate and then observed how it influenced the leak detection. When the leak valve was completely closed, the measured flow rate was 0 L/min, which corresponded to no leak on the pipeline. Opening the valve by quarter, half, and fully corresponded to leaks with flow rates of 7 L/min, 15 L/min, and 30 L/min, respectively, at a measured average pressure of 100 kPa.

To obtain the magnitude of the vibration in the X-, Y- and Z-axis, the measured acceleration in each of the 3D direction were subtracted from the zero-g offset value (the acceleration in that direction when there is no motion) and the result squared. They were summed up and the square root of the sum of the squares of the actual acceleration in each of the direction represented the resultant pipe surface acceleration. Given that X_{mea}, Y_{mea}, and Z_{mea} represent the acceleration measured in the X, Y, and Z direction, respectively and X_{0g}, Y_{0g}, and Z_{0g} represent the zero-g offset acceleration in the X, Y, and Z direction, respectively. The actual acceleration in all three directions (X_{act}, Y_{act}, and Z_{act}) were derived by subtracting the zero-g offset acceleration from the measured acceleration.

$$X_{act} = X_{mea} - X_{0g} \tag{6.1}$$

$$Y_{act} = Y_{mea} - Y_{0g} \tag{6.2}$$

$$Z_{act} = Z_{mea} - Z_{0g} \tag{6.3}$$

The magnitude of the resultant acceleration, A, caused by a leak is given by:

$$A = \sqrt{X_{act}^2 + Y_{act}^2 + Z_{act}^2} \tag{6.4}$$

Figure 6.4 represents the resultant acceleration for the various leak distances and leak sizes. For each leak distance and leak size, 25000 samples of vibration data were collected for a time duration of 27 seconds.

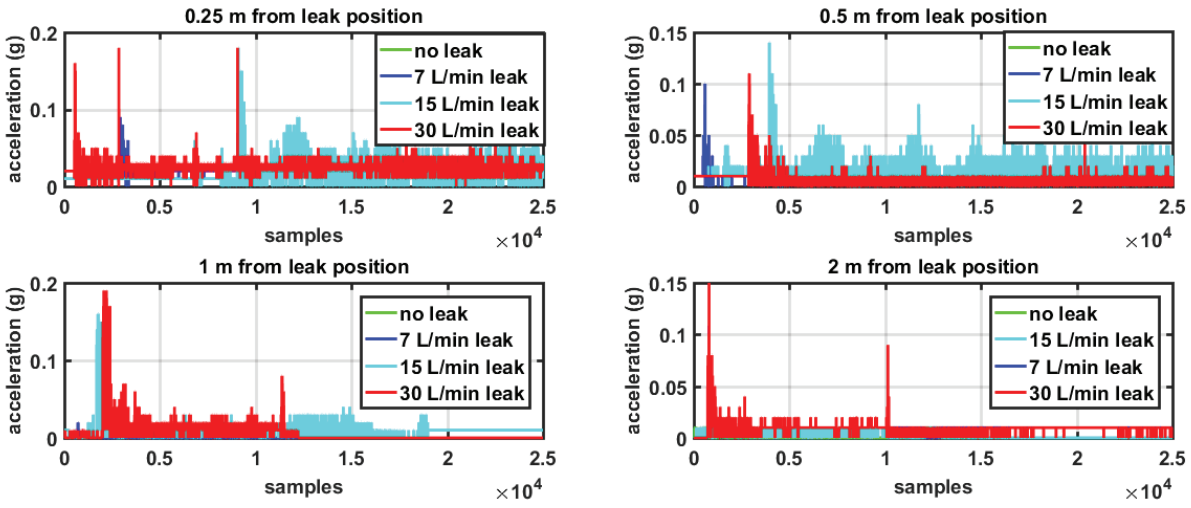

Figure 6. 4: Leak characterization

The results revealed that as the distance from the leak position increases, there exist no significant difference between the leak event data and the no leak event data. As shown in Figure 6.4, there was a significant difference between a leak event and a no leak event for distances of 0.25 m, 0.5 m, and 1 m from the leak position. The results also revealed that the amplitudes of the vibration at a distance of 0.25 m are much higher compared to the others. The amplitude decreases as the distance from the leak position increases. This is because the vibration signals are attenuated as the move away from the leak position, resulting in a decrease in their amplitude. At some distance far away from the leak position, the vibration signals can no longer be detected [46], [54], [55], [153]. From the data collected, it shows that at a distance of 2 m from the leak position, it becomes difficult to separate leak event data from no leak event data when compared to the other leak distances. Additionally, the results revealed that for each of the leak distances, there is a significant difference in the magnitude of the measured acceleration for leak sizes of 15 L/min and 30 L/min when compared with no leak. However, there is no significant difference between a leak of 7 L/min and a no leak event. This means that the accelerometer can significantly differentiate a leak from a no leak for leak sizes starting from 15 L/min. This implies that only fast leaks can be detected by the accelerometer. This observation is in agreement with those reported by [45]. From the results presented above, we can confidently say that by placing the accelerometer sensor 1 m from the leak position, we can significantly detect most of the leaks having sizes ranging from 15 L/min to 30 L/min. These results are consistent with previous studies [46], [54], [58], [125], [126].

Nonetheless, comparing our result with those of [58], we see that there is a difference in the maximum distance that the accelerometer can be placed from the leak position to effectively detect leaks. According to Ismail et al. [58], the maximum distance for effective leak detection derived using MPU6050 (which has similar characteristics as the LSM9DS1), was 0.5 m. However, our results revealed that we can effectively detect leaks at a distance of 1 m from the leak position using the LSM9DS1 accelerometer. The difference in the results can be explained by the fact that in our study we configured the accelerometer to the ±2 g sensing range which has more sensitivity (0.061 mg/LSB) whereas in [58], the authors configured the accelerometer to the ±16 g sensing range which has a lower sensitivity (0.732 mg/LSB). This means that to increase the sensor spacing and effectively detect leak in plastic pipes, it is necessary to configure the MEMS accelerometers to the lowest sensing range, since the magnitude of the vibrations on plastic pipes is low.

To obtain the baseline value, 86400 samples of acceleration data were collected each day for a period of 3 days and an average was computed to obtain the baseline value when there is no leak on the pipeline. Figure 6.5 represents the acceleration data collected for one day. By performing an average of the acceleration data collected for 3 days, we obtained the value 1.00912 g. This led us to use the value of 1.01 g as the baseline value for leak detection. Thus, a leak alarm triggered each time the difference between the estimated pipe surface acceleration and the baseline value exceeds the threshold value (δ) by 0.01 g.

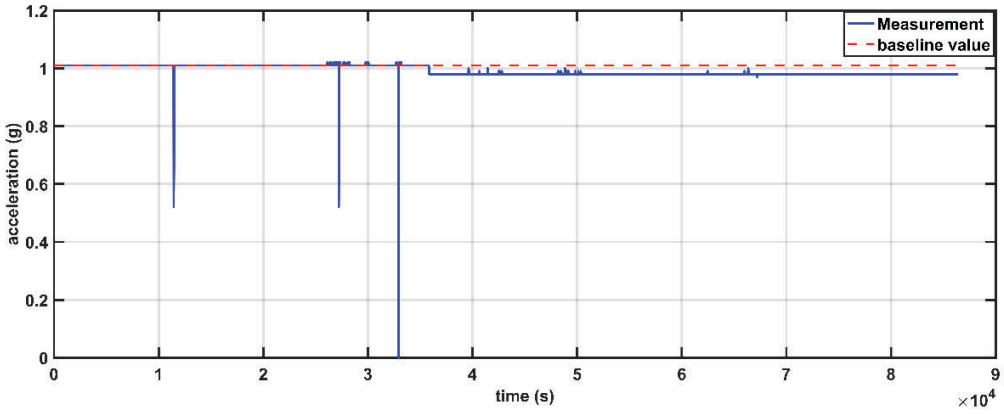

Figure 6. 5: Baseline acceleration value

In the next section, we present the simulation and laboratory results of the performance evaluation of the selected DKF algorithms.

6.3 Performance Evaluation

To evaluate the leak detection performance of the selected DKF algorithms, simulations were performed on CupCarbon using acceleration data collected from the field. The performance results of the selected DKF algorithms obtained from simulations were validated by results obtained from physical experiments on the laboratory testbed. Thus, we start this section by presenting the performance results obtained from simulations and then end the section by presenting and discussing the results obtained from physical experiments for validation purposes. This will partially answer the question of which DKF algorithm is better and well suited for application in a fully distributed leak detection solution in WWPM systems using low-cost MEMS accelerometers.

6.3.1 Performance Metrics

We used the performance metrics (sensitivity, specificity, and accuracy) developed in section 4.5 of chapter 4 to compare the leak detection performance of the three selected DKF algorithms. We used this metrics because of the following reasons:

- We needed to know with certainty which DKF algorithm was more reliable and this could not be easily visualized by using plots of the estimated acceleration.
- Recent WWPM studies (especially those implementing machine learning techniques) use this metric to evaluate the performance of their solution. Thus, this can enable us to easily compare the results of our proposed solution with those already existing in the literature.
- These metrics are commonly used in the industry to evaluate the performance of leak detection systems used in computational pipeline monitoring [195], [196].

6.3.2 Presentation and Discussion of Simulation Results

Figure 6.6 depicts the RMSE of the three selected DKF algorithms. For each DKF algorithm, the RMSE of both sensor nodes S1 and S2 are presented on the same plot.

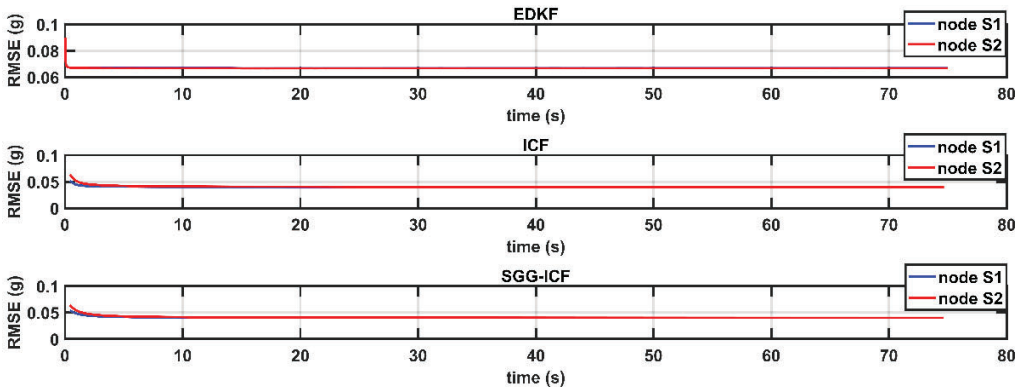

Figure 6. 6: RMSE of the selected DKF algorithms

From Figure 6.6, we see that there is no significant difference between RMSE of both sensor nodes S1 and S2 for all the DKF algorithms. The results in Figure 6.6 showed that a difference in the RMSE value of both sensor nodes S1 and S2 only occurred for the cases of ICF and SGG-ICF at the beginning of the simulation. However, this difference becomes insignificant with time as the RMSE values of both sensor nodes converge to the same value. This implies that all the three DKF algorithms compute consistent estimates and thus maintain local consistency in the estimates of neighbouring sensor nodes. This agrees with the results published in [59] , which showed that all the three DKF algorithms achieved local consistency when applied in a low-cost sensor network target tracking application. The property of local consistency is important for ensuring high reliability and reducing the false alarm rate of a WWPM system implementing DKF, given that it will prevent contradictory outputs from neighbouring sensor nodes as we saw in Section 5.2.

To compare the estimation accuracy of the three DKFs, Figure 6.7 depicts the RMSE of sensor node S1 for all three DKF algorithms. From Figure 6.7 we see that the RMSE of EDKF converges to 0.0667 while the RMSE value of SGG-ICF is slightly greater than that of ICF at the beginning and they both converge to the value 0.0397 with time. Also, it can be seen that ICF and SGG-ICF have lower RMSE values and can provide better estimation accuracy compared to EDKF. These results are also consistent with the results of He et al. [59]. Thus, we expect the leak detection performance of ICF and SGG-ICF to be higher than that of EDKF. To further evaluate the performance of the selected DKF algorithms, we carried out simulations on the two-node linear

WSN presented in sub-section 6.1.2 using acceleration data collected from the field. The results of the performance of the selected DKF algorithms from simulations are depicted in Figure 6.8.

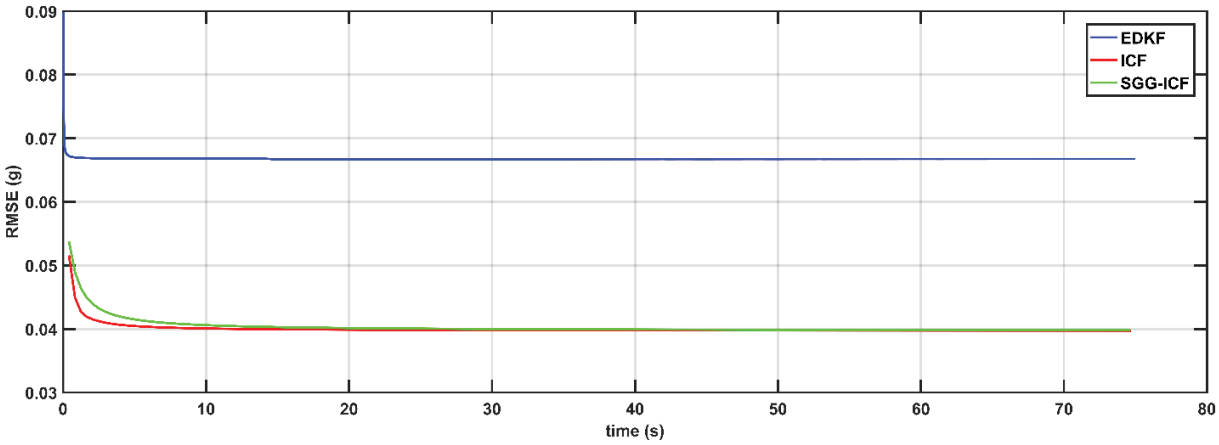

Figure 6. 7: Comparison of RMSE values of sensor node S1 for the selected DKF algorithms

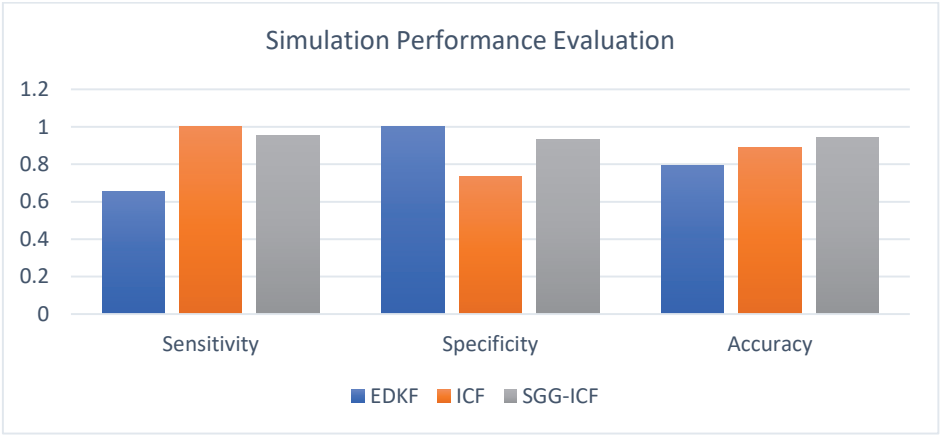

Figure 6. 8: Performance evaluation result obtained from simulations

From Figure 6.8, it can be seen that ICF and SGG-ICF have leak sensitivities which are significantly higher compared to that of the EDKF. ICF has the highest sensitivity (100%), followed by SGG-ICF with a sensitivity of 95% and finally the EDKF with a sensitivity of 65%. As shown in Figure 6.7, ICF had the lowest RMSE value and this explains why it has the highest sensitivity in Figure 6.8. However, the overall accuracy of the DKF algorithms revealed that SGG-ICF has the highest accuracy (94%) followed by ICF (88%), and lastly EDKF (79%). This goes further to support the conclusions of [59], which stated that SGG-ICF is well suited for distributed

state estimation in low-cost sensor networks as it provides more flexibility and strikes a balance between estimation accuracy and communication burden. However, based on the claim by Chan et al. [100], that accuracy may not be an optimal metric for evaluating the performance of a leak detection system as it is dependent on class proportions, we cannot at this point state that SGG-ICF is better. We will need further experiments to come to the conclusion of which DKF algorithm provides more reliable leak detection.

The presented simulation results imply that SGG-ICF and ICF will provide more reliable leak detection compared to EDKF. This is because in ICF and SGG-ICF, the sensor nodes exchange their local information multiples times between measurement updates whereas in EDKF, the sensor nodes communicate with their neighbours at most one time in between measurement updates. In addition, the event-triggered-commutation attribute of EDKF (which allows neighbouring nodes to approximate the local information pairs of their neighbours and not to communicate when the difference between the predicted state and the last transmitted state is below a defined threshold) reduces its estimation accuracy. However, this attribute makes the EDKF to have a lower communication requirement unlike SGG-ICF and ICF that have higher communication requirements which will eventually lead to high power consumption. To validate these simulation results, we present in the next sub-section the results of the performance of the three algorithms obtained from experiments conducted on the laboratory testbed.

6.3.3 Presentation and Discussion of Experimental Results

In this section, we present and discuss the results (Figure 6.9) obtained from the laboratory experiment scenarios described in sub-section 6.1.3.2 in order to validate the simulation results presented in sub-section 6.3.2.

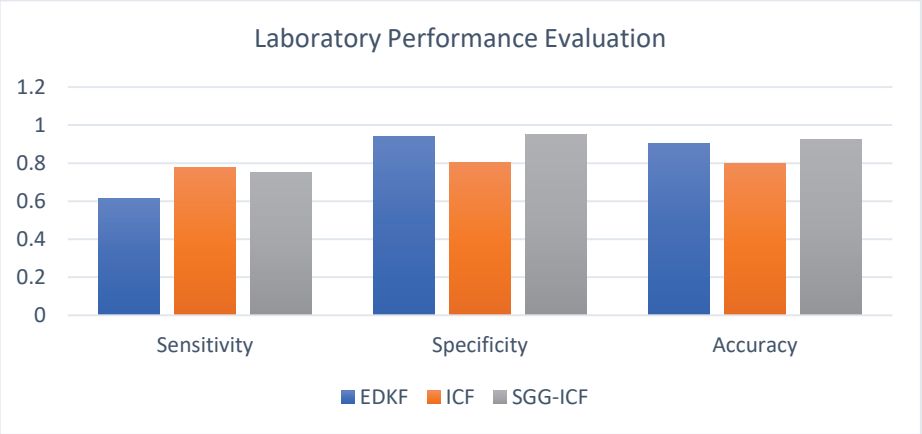

Figure 6. 9: Performance evaluation result obtained from laboratory experiments

Figure 6.9 represents the values of the performance metrics obtained for each of the DKF algorithms during the laboratory experiments. From Figure 6.9, the leak sensitivities are 61%, 77%, and 75% for EDKF, ICF, and SGG-ICF, respectively. From the sensitivity results, it can be seen that ICF detected most of the leak events that occurred and missed to detect fewer leak events compared to SGG-ICF and EDKF. EDKF with the lowest sensitivity of 61%, failed to detect 39% of the leak events that occurred on the pipeline, causing it to have the highest miss detection rate (MDR). This means that a high proportion of actual leaks will go unnoticed in the case of EDKF compared to the other algorithms. In terms of specificity, SGG-ICF is highest with 95% followed by EDKF with 93% and lastly ICF with 80%. From the specificity results, it can be seen that SGG-ICF correctly recognized most of the no leak events and generated fewer false alarms compared to ICF and EDKF. ICF with the lowest specificity of 80% declared 20% of no leak events as leak events causing it to have the highest false alarm rate (FAR). For a good leak detection system, it is better for the FAR to be higher than the MDR because the false alarms generated by the leak detection system can be ignored without it affecting the Non-Revenue Water (NRW). However, a higher MDR has an adverse effect on the NRW as it represents true leaks which occur on the WDN and are undetected but actually lead to water losses and a high NRW.

Knowing that sensitivity is a measure of how well the leak detection system detects true leak events while specificity is a measure of how well the system recognizes no leak events on the pipeline, this means that the DKF algorithm that has the highest sensitivity and specificity values is more

reliable. This combined effect of sensitivity and specificity is captured by the accuracy metric. From Figure 6.9, SGG-ICF has the highest accuracy (92%), followed by EDKF with 90%, and lastly ICF with 80%. This means that SGG-ICF is more reliable for leak detection compared to ICF and EDKF. This result is consistent with that obtained from simulations.

6.3.4 Comparison of Simulation and Experimental Results

In this sub-section, we compare the results obtained from laboratory experiments with the simulation results.

Figure 6.10 depicts a summary of the comparison of the performance evaluation results obtained from both simulations and laboratory experiments, Figure 6.11 presents the error between the simulation and laboratory results categorized by DKF algorithm, and Figure 6.12 presents the error between the simulation and laboratory results categorized by performance metric. The laboratory results are taking as the reference to compute the error.

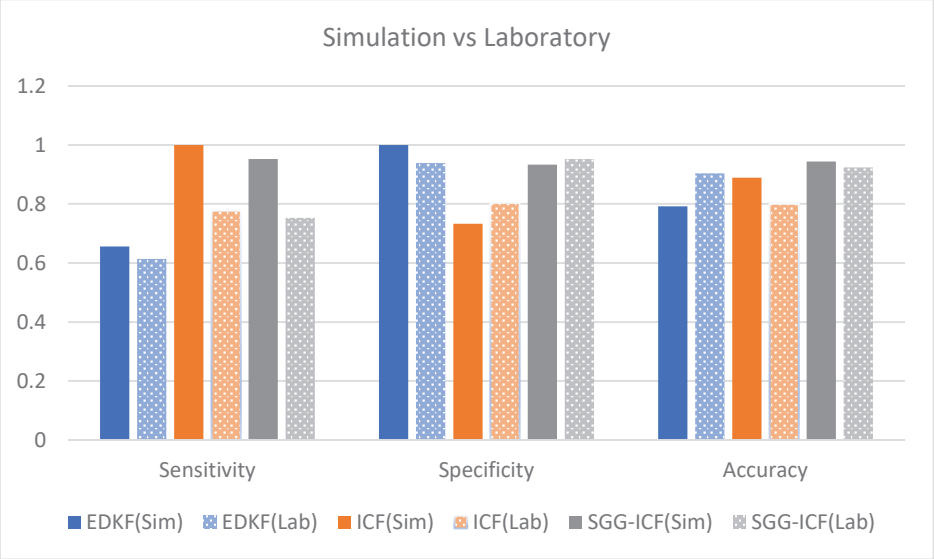

Figure 6. 10: Comparison of simulation and experimental performance values

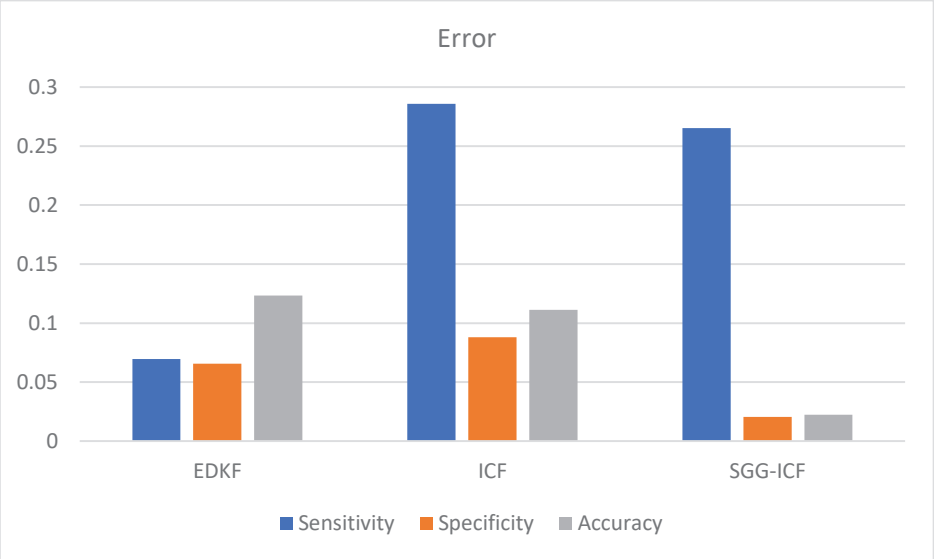

Figure 6. 11: Error categorized by DKF algorithm

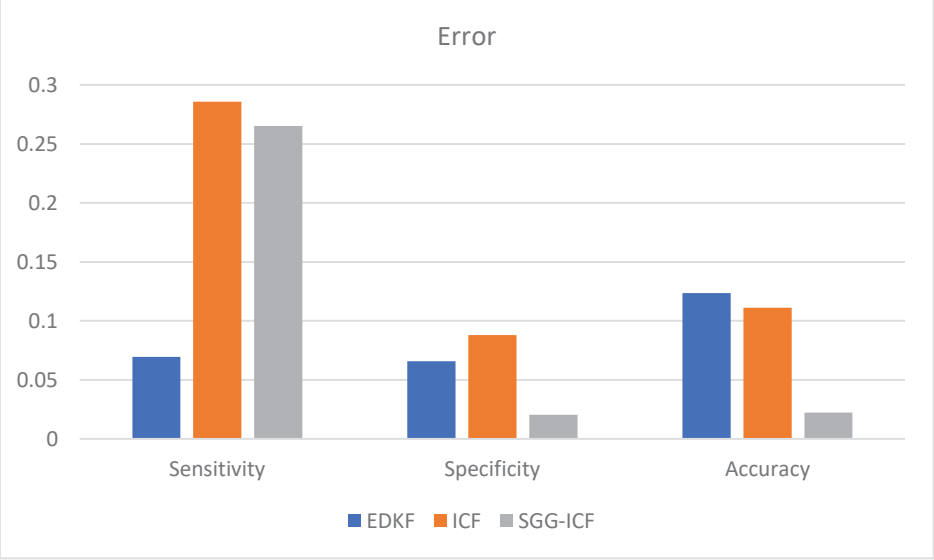

Figure 6. 12: Error categorized by performance metric

Comparing the results of the sensitivities obtained from simulations with those obtained from the laboratory experiments (Figure 6.10), we realize that there is a general decrease in the sensitivity obtained from the laboratory experiments when compared with those obtained from simulations. This decrease can be explained by the existence of packet loss during communication between neighbouring sensor nodes in the physical experiments which is absent in the simulations. We see that there is no significant difference in the sensitivity of EDKF obtained from the laboratory

experiments (61%) when compared to those obtained from simulations (65%). However, there are significant differences in the sensitivities of ICF and SGG-ICF obtained from simulations and laboratory experiments. For ICF and SGG-ICF, the sensitivities are 77% and 75%, respectively from laboratory experiments as compared to 100% and 95%, respectively, recorded from simulations. This result implies that ICF and SGG-ICF are greatly affected by packet loss compared to EDKF. This can be attributed to the high communication requirement of ICF and SGG-ICF (which involves large amounts of exchanges between neighbouring sensor nodes) and the highly unreliable wireless links in low-cost WSNs. For EDKF, its diffusion property alongside with its event-triggered nature drastically reduces the number of exchanges between neighbouring sensor nodes and thus reduces the packet loss rate. We observed that in the physical experimentation of the EDKF, the packet loss rate was very low (<5%). This makes EDKF very appropriate for real-time application in systems where the dynamics of the system is changing fast. For ICF and SGG-ICF, which require multiple communications rounds between successive measurement updates to achieve excellent estimation accuracy, it is evident that their overall estimation accuracy depends on the packet loss rate. However, given that we are dealing with low-cost sensor networks where the communication links are unreliable, this increase in the number of data exchange between neighbouring sensor nodes will increase the likelihood of packets being lost. We observed in the physical experiments that out of 5 data exchanges that occurred between neighbouring sensor nodes during the information fusion stage, only 60% of the transmitted packets were received successfully, meaning that only three out of five messages transmitted were successfully received. This explains the significant difference in the sensitivities of ICF and SGG-ICF obtained from simulations and laboratory experiments. The results in Figure 6.11 confirm that EDKF is the DKF algorithm least affected by packet loss, followed by SGG-ICF and lastly ICF. Our results agrees with the proposition of He et al. [59], which suggested the use of diffusion-based DKF algorithms in situations where the communication resources are limited. Furthermore, it can be seen from Figure 6.12 that sensitivity is the performance metric most affected by packet loss. This can be explained by the fact that the occurrence of a leak on the pipeline leads to a sudden increase in the measured pipe surface acceleration, which results in an estimated acceleration which is significantly different from the previously estimated acceleration when there was no leakage on the pipeline. The DKF algorithm is required to react fast in order to capture this

sudden change. As such, any delay resulting from packet loss and retransmission will minimize the chances of detecting this sudden increase in the pipe surface acceleration. However, the response time for ICF and SGG-ICF is slow since they have to involve in numerous communication rounds between measurements. The loss of packets due to the unreliable wireless links in low-cost WSNs further worsens the estimation accuracy. Thus, to achieve high sensitivity, it is required that measurements be treated in a timely manner as they are obtained.

The results in Figure 6.10 reveal that there is no significant difference between the specificity obtained from simulations and that obtained from physical experiments. From the results, we see that SGG-ICF has the highest specificity in the physical experiments as opposed to EDKF which has the highest specificity from the simulation results. Generally, from the simulation and physical experiments results, we observed that the specificities of the DKF algorithms are high. This means that there is a low likelihood of an alarm being triggered when there is no real occurrence of a leak on the pipeline and this increases the reliability of the leak detection system.

In terms of accuracy, SGG-ICF still has the highest accuracy (92%) which is slightly lower than that obtained from simulations (94%). In the same light, the accuracy of ICF obtained from laboratory experiments (80%) is significantly lower than that obtained from simulations (88%). However, we noticed that the accuracy of EDKF obtained from laboratory experiments (90%) is significantly greater than that obtained from simulations (79%). We also noticed that the accuracy of ICF obtained from laboratory experiments is lower than that of EDKF, which contradicts the results obtained from simulations, where EDKF has the lowest accuracy value. This results also confirm that accuracy is not a perfect metric for evaluating the performance of leak detection techniques due to its biasedness, based on its dependence on class proportions as earlier stated in [100]. Thus, accuracy is not a preferred metric when dealing with skewed datasets.

In the next section, we will evaluate the power consumption of the selected DKFs via simulations and physical experiments. This will permit us determine which DKF algorithm is optimal, taking both performance and power consumption into consideration.

6.4 Power Consumption Evaluation

In this section we evaluate the power consumption of the three selected DKF algorithms by presenting and discussing the results obtained from both simulations and physical measurements. We start by presenting the results from simulations in sub-section 6.4.1 then validate them with results from laboratory experiments in sub-section 6.4.2.

6.4.1 Results from Simulations

We conducted the simulations on the two-node linear WSN presented in sub-section 6.1.2. For each of the selected DKF algorithm that was implemented on the two-node linear WSN, we ran the simulation for a period of 1 day (86,400 s). Figure 6.13 depicts the energy profile of sensor node S2 for all the three selected DKF implementations.

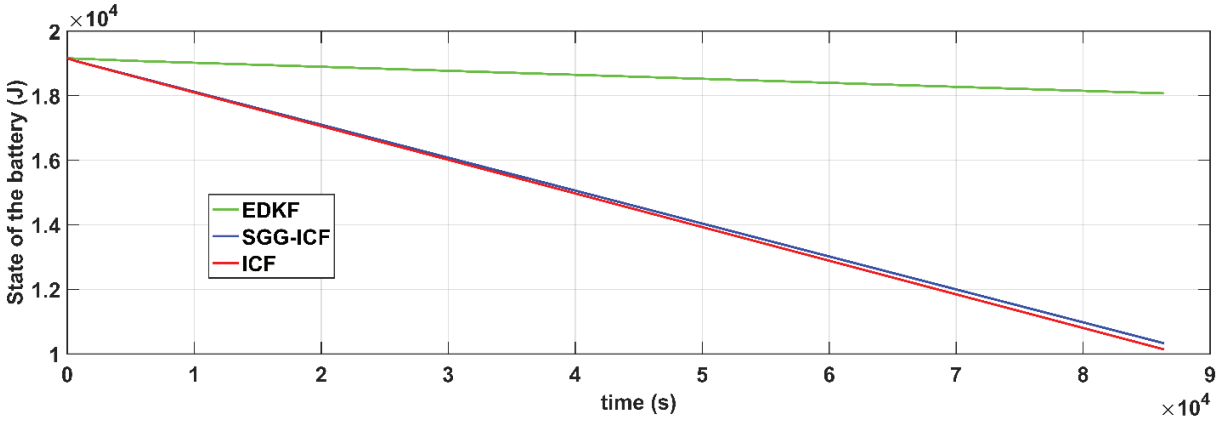

Figure 6. 13: Comparison of the energy consumption of the selected DKF algorithms

From Figure 6.13, it is clear that EDKF has the lowest energy consumption, while ICF consumes more battery energy compared to the other DKF algorithms. We also observed that the energy consumption of ICF and SGG-ICF were close. This is because both of them involved in the same number of consensus/gossip iterations. From the results displayed in Figure 6.13, we see at time t = 80,000 s, the battery energy of sensor node S2 has not crossed the 1.8×10^4 J level for the EDKF implementation, whereas the battery energy of S2 had crossed the 1.8×10^4 J level at time t = 10,000 s for both ICF and SGG-ICF. Summarily, the results revealed that for a period of 1 day (t = 86,400 s), sensor node S2 had exhausted 5.7%, 46.1%, and 47.1% of its total battery energy in the EDKF, SGG-ICF, and ICF implementations, respectively. Thus, we see that the energy consumption of SGG-ICF and ICF are over 8 times greater than the energy consumption of EDKF.

The results revealed that EDKF is a more energy efficient solution and will provide a longer WWPM lifetime when compared to ICF and SGG-ICF.

The next sub-section presents physical experiments that measured the power consumption of the DKF algorithms and the comparison results with those obtained from simulations.

6.4.2 Results from Laboratory Experiments

For each of the DKF implementation, we measured the current and voltage of the nodes. Figure 6.14 represents the power profile of sensor S2 in the case of the EDKF, ICF, and SGG-ICF implementations.

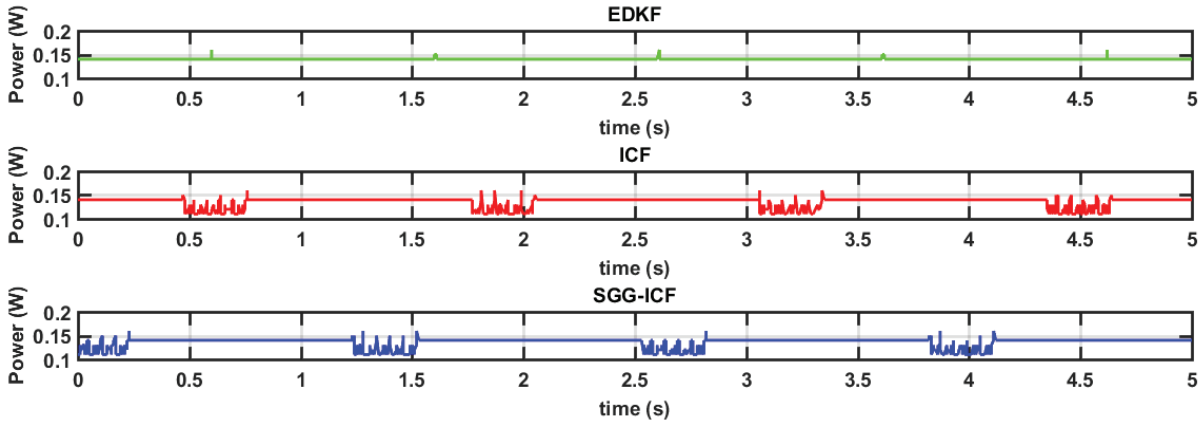

Figure 6. 14: Power profile of the three DKF algorithms

The average power consumed by the sensor node S2 for a period 1 hour was found to be 135.616 mW for ICF and SGG-ICF, while it was 140.064 mW for EDKF. From these results we that the lifespan of sensor node S2 when powered by the 2000 mAh capacity Li-Po battery will be approximately 54 hours for ICF and SGG-ICF, and approximately 52 hours for EDKF. Thus, the lifespan for the sensor node is approximately 2 days for all the DKF implementations. We also observed that the difference in power consumption for EDKF when compared to ICF and SGG-ICF was not as significant as what we had obtained from simulations. This can be explained by the fact that CupCarbon simulator only models the energy consumption of the communication unit. The energy consumption of the processing unit and sensing unit of the sensor node are not taking into consideration. We also observed something unexpected from the power measurement results obtained from the physical experiments. As can be seen from Figure 6.14, the power consumption

of EDKF is higher than that of ICF and SGG-ICF. This is inconsistent with the results obtained from simulations and also from a theoretical perspective. For verification, we powered the sensor node for a period of 1 day using the Li-Po battery and monitored the rate of discharge of the battery's supply voltage. Figure 6.15 illustrates the state of the sensor node battery's voltage over a period of 1 day for each of the DKFs implementation.

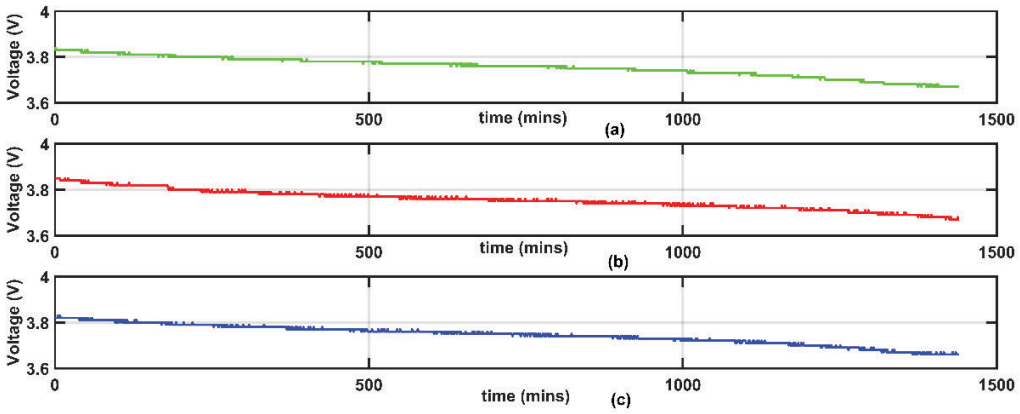

Figure 6. 15: State of charge of sensor node's battery for the selected DKF implementations: (a) EDKF, (b) ICF, (c) SGG-ICF

We then computed the state of charge (SOC) of the battery using equation (4.35).

Table 6. 2: Battery energy consumption

	Battery Voltage (V)		SOC (%)		Battery energy usage (%)
Time (mins)	0	1440	0	1440	
EDKF	3.83	3.67	67.32	26.61	40.71
ICF	3.85	3.67	71.14	26.61	44.53
SGG-ICF	3.82	3.66	65.26	23.92	41.34

Table 6.2 shows the battery energy consumption for the three selected DKF algorithms. As can be seen, EDKF has the lowest energy consumption (40.71%), followed by SGG-ICF (41.34%), and lastly ICF (44.53%). These results are consistent with those obtained from simulations. The results also show that there is no significant difference in the energy consumption of ICF and SGG-ICF, which also consistent with the result obtained from simulation. However, the difference between the energy consumption of EDKF and ICF is not as significant was revealed by the simulation results. From simulations, the energy consumption of ICF was 8 times higher than that of EDKF

whereas the results from the laboratory experiments revealed that the energy consumption of ICF is less than 2 times the energy consumption of EDKF. One reason for this great difference can be explained by the fact that the CupCabon simulator does not model the energy consumption of the microcontroller and the sensor.

6.5 Optimal DKF Algorithm for Leak Detection in WWPM Systems using Low-cost MEMS Accelerometers

From the results of leak detection performance and power consumption presented, a compromise between leak detection performance and energy efficiency shows that EDKF is more optimal for a fully distributed leak detection solution in WWPM when compared to ICF and SGG-ICF, since its estimation performance is less affected by packet loss. Rather than implementing ICF or SGG-ICF which have higher sensitivities than EDKF, but have a high communication requirement (which has an adverse impact on the power consumption), we propose that the sensitivity of a WWPM solution implementing EDKF can be improved by using machine learning (ML) techniques at the decision level [45]. Thus, EDKF can be used for filtering at the feature extraction level to provide more accurate features which are then classified by a ML algorithm at the decision level to provide accurate leak detection. Since this approach involves more computation than communication, it will likely reduce the energy consumption at the sensor nodes, making it a good candidate for implementation in battery-powered sensor nodes used in WWPM, given that it is more energy-aware.

Lastly, from the results displayed in Figure 6.14 and Table 6.2, it can be seen that the lifespan of the sensor node is short. This is because in these implementations, the ESP32 main core is operational throughout and as we saw in section 5.3 of chapter 5, the ESP32 is a very power-hungry chip when operating in the modem sleep mode, given that it is a middle-end device. This huge power consumption is not appropriate for a WWPM system as it is required to operate for long periods without the need to recharge or replace the batteries. Thus, reducing the power consumption necessitates that the ESP32 be configured as a low-end device by putting it most of the time in the deep sleep mode. However, to ensure real-time leak detection, it is necessary that the ESP32 awakes from sleep when there is a leak event to perform the processing tasks required

to reliably detect the leak. Thus, the challenge is how to achieve low-power consumption without compromising the real-time performance of the WWPM system. We tackle this by proposing hierarchical sensing and duty cycling to achieve both low-power consumption and real-time detection. In the next section, we present how we reduced the sensor node power consumption via duty cycling and hierarchical sensing, while maintaining real-time leak detection.

6.6 Power consumption Reduction via Duty Cycling and Hierarchical Sensing

In this section, we present and discuss the results of the implementation of hierarchical sensing and duty cycling at the node level to achieve both low-power consumption and real-time monitoring.

The operation at each sensor node is as follows:

- The ADXL344 is in the active mode where it continuously monitors the pipeline to detect an activity. An activity (a leak event) is detected once the measured acceleration is above a predefined threshold value that is stored in the activity register of the ADXL344 accelerometer. Once an activity is detected, an external interrupt is sent to wake up the other components of the sensor node.
- When no activity is detected by the ADXL344 accelerometer, the ESP32 stays in the deep sleep mode with the ULP coprocessor active, while the nRF24L01+ transceiver and the LSM9DS1 accelerometer both remain in the power-down mode.
- Once an activity is detected, the LSM9DS1 wakes up and collects more accurate measurements. The measurements are then processed by the ESP32 main core by running the DKF algorithm. The nRF24L01+ is used to communicate the local estimates of the sensor node to its direct neighbours to enable fusion.
- Once the fusion of local estimates from neighbouring nodes has been performed, the computed estimate is then compared with the baseline value. If the final estimate exceeds the baseline value by some threshold value, then a leak alarm is triggered and the node goes back to sleep. Otherwise, no leak alarm is triggered and the node goes back to sleep.

Figure 6.16 depicts the flow of operations in our proposed fully distributed, real-time and low-power leak detection solution for WWPM using low-cost MEMS accelerometers.

For illustration purposes, we implemented the proposed solution on the laboratory testbed and evaluated the power consumption of the ICF algorithm (since it has the highest power consumption from previous experiments in section 6.4). Figure 6.17 illustrates the current profile of sensor node S2 and Figure 6.18 depicts the battery voltage discharge over a period of 1 day.

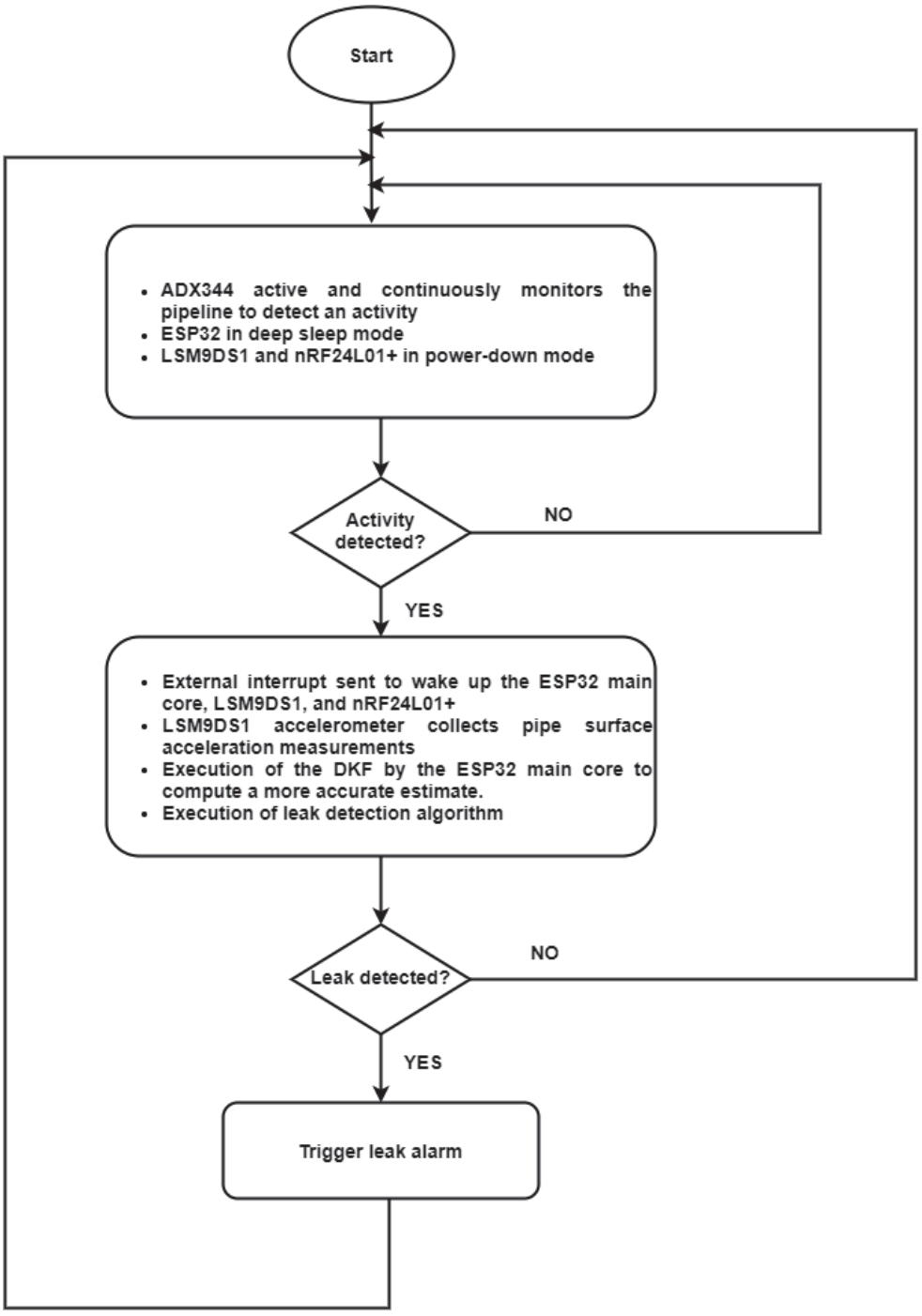

Figure 6. 16: Proposed fully distributed, real-time and low-power leak detection solution for WWPM using low-cost MEMS accelerometers

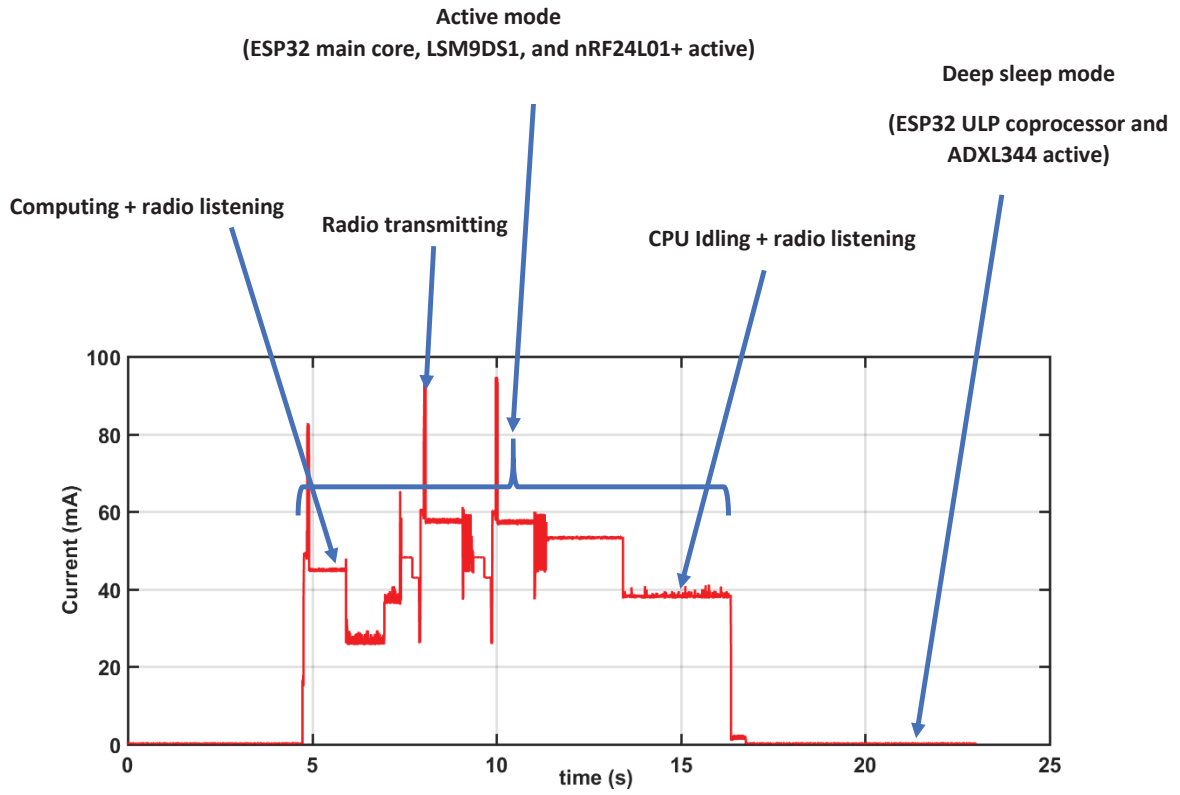

Figure 6. 17: Current profile of sensor node implementing duty cycling and hierarchical sensing

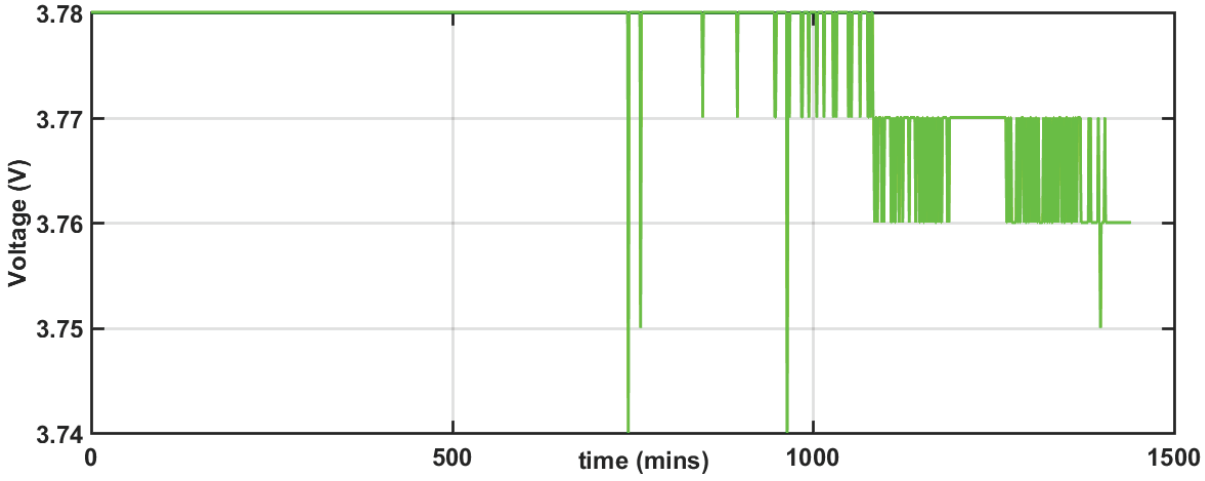

Figure 6. 18: State of charge of sensor node's battery when duty cycling and hierarchical were implemented.

The results in Figure 6.17 revealed a current consumption of 300 µA (0.3 mA) when the node is in the deep sleep mode. This is significantly close to the theoretical datasheet value of 290 µA

(150 µA for ESP32 ULP coprocessor and 140 µA for ADXL344 measurement mode supply current). From the results, it can be seen that there is a significant decrease in the power consumption of the sensor node. When there is no leak on the pipeline, the sensor node consumed a current as low as 0.3 mA when duty cycling and hierarchical sensing were implemented compared to 31.8 mA when duty cycling and hierarchical sensing were not implemented. The results in Figure 6.18 showed that the battery voltage dropped from 3.78 V to 3.76 V for a period of 1 day when both duty cycling and hierarchical sensing were implemented at the sensor node level. This corresponded to a battery energy consumption of 5.03% compared to 44.53% when duty cycling and hierarchical sensing were not implemented. Thus, it can be seen that the implementation of duty cycling at the sensor node level can lead to an increase the sensor node's lifetime by a factor of 8. This is because the node stays in the ultra-low power state (where its current consumption is 0.3 mA) whenever there is no leak on the pipeline. Finally, the results presented in this section demonstrates that our proposed solution can achieve reliable real-time leak detection while at the same time preserve the lifespan of the WWPM system.

6.7 Summary

In this chapter, we carried out leak characterization on the laboratory and the results obtained revealed that leak sizes ranging from 15 L/min could be effectively detected at a distance of 1 m when the accelerometer was configured to the ±2g sensing range. We also performed simulations and physical experiments to evaluate the leak detection performance and the power consumption of the selected DKF algorithms. The results from simulations and laboratory experiments revealed that ICF had the highest leak sensitivity while EDKF had the lowest leak sensitivity. The results of the leak detection performance for EDKF derived from simulations were closed to that obtained from the laboratory experiments. However, there was a significant difference between the leak detection performance of ICF obtained from simulations and that derived from the laboratory experiments. The difference was explained by the fact that there was loss of packets in the physical experiments during communications between sensor nodes that were not considered during simulations. Besides, the power consumption results obtained from simulations revealed that the power consumption of the EDKF was significantly lower than that of ICF and SGG-ICF. Thus, by performing a trade-off between leak detection performance and energy efficiency, it could be seen

that EDKF is optimal in a practical implementation compared to ICF and SGG-ICF, since its estimation performance is less affected by packet loss. In addition, its low communication burden reduces the power consumption of the sensor nodes. This makes EDKF a good candidate for implementation in battery-powered sensor nodes. Conclusively, a trade-off between energy consumption and leak detection performance revealed that EDKF is a better solution for the implementation of real-time leak detection in battery-powered WWPM solutions. Finally, we demonstrated how the power consumption of our proposed WWPM solution could be reduced by implementing duty cycling and hierarchical sensing at the sensor node level. The results revealed a decrease in the power consumption by a factor of 8. This revealed that our proposed solution can achieve reliable real-time leak detection and at the same time preserve the lifespan of the WWPM system.

Chapter 7

Conclusion and Outlook

7.1 Conclusions

In this thesis, we were interested in proposing a real-time, reliable, low-power, and fully distributed solution for leak detection in WWPM systems using low-cost MEMS accelerometers. Our main goal was to achieve an increase in measurement accuracy and network lifetime via a distributed approach. To achieve our goal, we had five specific objectives to attain (objectives 1 to 5), which we realized in chapters 2 to 6, respectively. In this section, we state the conclusions drawn from the realization of the objectives.

In Chapter 2, it is shown that in centralized WWPM systems, there is high power consumption and degradation in the real-time performance due to the large number of multi-hop communications involved. Thus, distributed computing rather than centralized computing is required for achieving low-power and real-time leak detection in WWPM systems using low-cost non-intrusive sensors. Besides, the investigation of popular non-intrusive leak detection techniques in WWPM systems shows that low-cost MEMS accelerometers can be employed for detecting leaks in plastic pipes but still require additional signal processing in order to increase the leak detection accuracy. Signal processing techniques that process data straightforwardly in the time domain and that involve multi-sensor data fusion can increase the leak detection performance and lower the power consumption of WWPM systems. Implementing such algorithms at the sensor node level requires that the sensor nodes be low-power but have sufficient computing power.

In recent years, advances in microelectronics have led to the development of low-power microcontrollers (e.g., ESP32) with higher computational power and memory capacity. Chapter 3 reveals that by using the ESP32 more frequently in the ULP mode than in the modem sleep mode, the ESP32-based sensor node has a power consumption comparable to that of low-end devices used in battery-powered WSN applications but also possess sufficient computing resources to perform all the processing required for leak detection within the sensor node. Additionally, the ability of the ESP32 to monitor the voltage level of its battery supply without needing extra

circuitry gives the sensor nodes the capability of monitoring their energy consumption in real-time, making them energy-aware and adaptive.

The existence of a myriad of DKF algorithms and the need for thorough survey of DKF algorithms to isolate DKF algorithms with the potential for implementation in WSNs is performed in Chapter 4. Besides, the need for neighbouring sensor nodes to have consistent estimates hugely affects the leak detection accuracy of a fully distributed WWPM system using low-cost MEMS accelerometers. Thus, avoiding such controversies implies implementing only DKF algorithms that achieve consistent local estimates for leak detection in WWPM systems.

In chapter 5, the viability of a fully distributed DKF-based solution for leak detection in WWPM systems using low-cost sensors is demonstrated practically. A combined approach involving simulations and laboratory experiments is validated and a simulation model for the power consumption developed. The results established the importance of distributed data fusion and the pertinence of the distributed approach in improving the leak detection performance and preserving the WWPM system's lifetime. The impact of these results is not only limited to WWPM systems, but it can be extended to time-critical IoT applications. In this case, instead of shipping raw data to the cloud for processing and sending back the decisions, processing the raw data of a time-critical event monitoring application using the onboard computing resources of the sensor nodes can reduce the latency.

In chapter 6, the evaluation and comparison of the leak detection performance and power consumption of three selected DKF algorithms using the combined approach is demonstrated. From the simulation and laboratory experiment results, a trade-off between leak detection performance and energy efficiency reveals that diffusion-based DKF algorithms are optimal for practical implementation in battery-powered sensor nodes when compared to consensus-based and gossip-based DKF algorithms. Finally, implementing hierarchical sensing and duty cycling at the sensor node significantly decreases the energy wasted in idle listening. The results showed a reduction in the power consumption by a factor of 8. This implies that our proposed solution can achieve reliable real-time leak detection and at the same time preserve the lifespan of the WWPM system.

The conclusion of this thesis is that it is possible to implement a fully distributed DKF-based solution that increases the leak detection performance and lifespan of a WWPM system composed of a network of low-cost MEMS sensors. The original contribution to knowledge of this thesis is the proposition, demonstration and evaluation of this fully distributed DKF-based solution for reliable real-time leak detection in plastic WDNs using a WWPM system that employ low-cost MEMS accelerometers.

7.2 Limitations and Recommendations for Future Work

This study has some limitations based on certain assumptions that we made. In this section, we present the limitations of our study and also suggest future directions of research that can be explored.

Limitation 1: We assumed there were no vibrations on the pipeline due to legitimate customer demands, i.e., vibrations resulting from the opening/closing of taps at the customer premises. In addition, the effects of pumps were neglected as we considered only cases where the distribution of water in the pipeline is by means of gravity. Furthermore, we considered above-the-ground pipelines in our laboratory experiments but most WDNs in developing countries are underground pipelines. Lastly, our study was limited to a two-node linear WSN.

Recommendation 1: Though results obtained from the laboratory testbed were satisfactory, extending our experiments to a field study that involves the deployment of a large-scale linear WSN on a real WDN with real-life conditions is suggested for future work. This is important because the simplistic nature of the laboratory WDN does not capture all the complications in a real WDN. Another interesting point is that most WWPM studies are limited to simulations and experiments on laboratory testbeds. Thus, extending experiments to real WDNs will contribute to the literature of WWPM.

Limitation 2: Another limitation of this study is the use of the thresholding method with a fixed baseline value for determining the presence or absence of a leak on the pipeline. Such a simplistic approach is more suitable in pipeline systems with a predictable operational characteristic. However, given that the operational conditions of real WDNs are not constant and might change

as a result of external conditions that are not related to a leak, it is required that the baseline value be able to vary.

Recommendation 2: We suggest the implementation of machine learning techniques at the decision step of the leak detection algorithm. This implies that once the DKF has been used at the feature extraction phase to estimate the pipe surface vibration, the value can then be passed to a trained classifier at the decision phase to accurately determine the existence of a leak or no leak on the pipeline.

Limitation 3: This study is only limited to leak detection. However, to have a complete WWPM system, it is necessary to incorporate leak localisation.

Recommendation 3: Explore the implementation of leak localisation techniques such as acoustic correlation analysis in future experiments.

Limitation 4: The CupCarbon simulator models only the energy consumption of the communication unit without considering the energy consumption of the sensing and processing units.

Recommendation 4: Given that CupCarbon simulator is open source, we suggest for future work the modification of its source code to include the energy consumption of the processing and sensing units. This will lead to simulated power profiles of the sensor nodes close to those obtained from physical measurements. Such simulated power profiles can then be used as an accurate first-hand assessment of the power consumption of distributed algorithms before physical implementation.

Résumé étendu (Extended Summary in French)

Chapitre 1 : Introduction

Ce chapitre présente les applications et les contraintes des réseaux de capteurs sans fil - Wireless Sensor Networks (WSN) - ainsi que la nécessité de passer à une architecture distribuée dans les applications WSN. Nous discutons brièvement de la surveillance des canalisations d'eau - Water Pipeline Monitoring (WPM) - en mettant l'accent sur les défis de la distribution de l'eau dans les pays en voie de développement et sur le nécessité d'avoir des systèmes fiables de détection des fuites pour la WPM. La motivation pour une solution de calcul distribué pour le WPM utilisant le WSN est également présentée. Les objectifs de la recherche et la contribution originale à la connaissance sont discutés, suivis par la portée de l'étude et l'organisation de la thèse.

1.1 Vers un paradigme distribué dans les réseaux de capteurs sans fil

1.1.1 Applications de WSN

Un réseau de capteurs sans fil (WSN) se compose de plusieurs nœuds intégrés avec des capacités de détection, de traitement et de communication sans fil, répartis sur une zone d'intérêt pour surveiller les conditions physiques ou environnementales [1]. Étant des systèmes spatialement distribués, les WSN exploitent la communication sans fil comme moyen de communication entre les nœuds. Cela les rend efficaces pour une myriade d'applications.

Les domaines d'application des WSN comprennent la surveillance géographique, la surveillance de l'habitat, le transport, les systèmes militaires, les processus commerciaux, la recherche sur le microclimat, les soins médicaux et autres [2]. Kandris et al. [3], dans une récente étude sur les applications du WSN, a classé les applications du WSN en six grandes catégories (militaire, environnementale, santé, flore et faune, industrielle et urbaine) en se basant sur les la nature de leur utilisation.

1.1.2 Contraintes des WSN

Un nœud de capteurs sans fil est généralement une unité matérielle de taille compacte qui acquiert les données souhaitées dans l'environnement et communique sans fil avec d'autres nœuds du réseau de manière à relayer les données brutes recueillies ou les informations extraites vers un puits de

données central. Ses principaux composants comprennent : une unité de communication, une unité de contrôle et de traitement, une unité de mémoire, une unité d'alimentation électrique et une unité de détection [4]. Ainsi, un WSN peut être composé de nœuds qui ont le potentiel de détecter, calculer et communiquer. Cependant, les nœuds de capteurs sans fil sont intrinsèquement limités en termes de ressources, ayant généralement une capacité de traitement, une capacité de stockage et une bande passante de communication limitées. Ces limitations sont en partie dues aux deux plus grandes contraintes, à savoir l'énergie limitée et la taille physique [5].

Dans un WSN à grande échelle, l'un des principaux objectifs est de parvenir à une faible consommation d'énergie, afin de permettre aux nœuds de capteurs d'être opérationnels pendant de longues périodes sans remplacer leur batterie, car les nœuds de capteurs dans ces applications sont généralement alimentés par une batterie et souvent inaccessibles physiquement. Au fil des ans, en raison des progrès technologiques, l'architecture matérielle des nœuds de capteurs sans fil a évolué, passant des nœuds de capteurs de première génération (par exemple Tmote Sky, MicaZ, Mica2, Micadot, etc.) qui utilisent des microcontrôleurs de 8 bits, aux nœuds de capteurs de deuxième génération (par exemple TelosB) qui intègrent des microprocesseurs. g. TelosB) qui intègrent des microcontrôleurs 16 bits tels que le MSP430, et enfin aux nœuds de capteurs de troisième génération qui utilisent des microcontrôleurs 32 bits (par exemple, des microcontrôleurs 32 bits basés sur ARM Cortex -M0/M0+/M3/M4/M7, ESP32 à double cœur et PIC32MZ). [6], [7]. Les nœuds de capteurs de premières et deuxièmes générations n'effectuaient que peu ou pas de traitement local en raison de la contrainte de leur puissance de calcul et de leur mémoire embarquée. Par contre, les nœuds de capteurs de troisième génération disposent d'une puissance de calcul et d'une mémoire embarquée plus importantes qui leur permettent de réaliser un traitement *in situ*.

1.1.3 Évolution vers une approche de calcul distribué dans les WSN

La plupart des applications de surveillance WSN reportée dans la littérature, notamment le WPM sont centralisées [8]–[10]. Ceci s'explique par le fait que les premiers nœuds de capteurs agissaient comme de simples collecteurs de données et relais sans fil à cause de leur faible puissance de calcul embarquée qui ne pouvait leur permettre que de capter et de communiquer des données plus ou

moins prétraitées. Cela conduit à la sous-utilisation de l'unité de traitement et à la surutilisation de l'unité de communication des nœuds de capteurs puisque leur rôle principal dans ces architectures centralisées est de collecter et de transmettre périodiquement des données à une station de base centrale intelligente, où tout le traitement est effectué pour détecter les comportements anormaux [8], [11], [12]. En outre, dans les applications de surveillance à grande échelle, la plupart des nœuds de capteurs sont géographiquement éloignés de la station de base et sont généralement alimentés par batterie [13]. Les inconvénients des WSN à architectures centralisées déployés dans les applications de surveillance à grande échelle incluent des besoins énormes en bande passante et une consommation d'énergie élevée. En effet, la transmission périodique de données brutes sur de longues distances, via de multiples sauts, vers la station de base entraîne un épuisement rapide de la batterie des nœuds de capteurs et réduit la durée de vie d'un WSN [12], [14], [15], car les recherches ont montré que l'unité de communication consomme la plus grande partie de l'énergie d'un nœud de capteurs [1], [17]. Il est donc évident que la communication consomme beaucoup plus d'énergie que le calcul. Pour réduire la consommation d'énergie des nœuds de capteurs dans un WSN, il est judicieux d'investir davantage dans le calcul au sein du WSN, autant que possible, afin d'économiser sur les coûts de communication. De la sorte en minimisant autant que possible la quantité et la portée des communications par une collaboration locale entre les nœuds de capteurs, on peut prolonger de manière significative la durée de vie d'un WSN. Enfin, les applications WSN centralisées présentent d'autres inconvénients, notamment une fiabilité et une robustesse limitées, un temps de réponse plus long, ainsi qu'un faible niveau de sécurité et de confidentialité des données [12], [14], [15], [18], [59]. Ces inconvénients des WSN centralisés ont conduit à une recherche active ces dernières années orientées vers le calcul distribué dans les WSN.

Récemment, certains travaux dans la littérature ont démontré via des simulations, la faisabilité du calcul distribué dans les WSN et ses promesses de gain de performance et de réduction de la consommation électrique [9], [12], [18], [20].C’est pourquoi, en effectuant davantage de calculs locaux, en limitant les échanges uniquement entre les nœuds voisins et en réduisant le nombre de messages à transmettre, le calcul distribué dans les WSN a le potentiel d'apporter une solution aux inconvénients de l'approche centralisée [18], [21], [22].

1.2 Surveillance des canalisations d'eau

1.2.1 Structure des systèmes d'approvisionnement en eau

L'eau est une nécessité de base pour la vie quotidienne. Elle est requise pour de nombreuses activités humaines telles que la boisson, l'irrigation des cultures, les activités récréatives, et pour l'accomplissement efficace de nombreux processus industriels [23]. Dans la plupart des communautés, le transport de l'eau par canalisation jusqu'aux utilisateurs semble être le moyen le plus économique [10], et consiste en des systèmes d'approvisionnement en eau comprenant deux parties principales: (1) Les conduites de transmission, qui sont des canalisations chargées de transporter l'eau vers les réservoirs et (2) les réseaux de distribution d'eau (RDE) , qui sont des canalisations et des raccordements de service pour distribuer l'eau aux utilisateurs. Ces infrastructures ne sont généralement pas totalement étanches, car même dans les réseaux de distribution d'eau les plus récents et les mieux construits, il existe un certain niveau de fuite et des ruptures occasionnelles de tuyaux, ce qui entraîne des pertes d'eau [24].

1.2.2 Les problèmes de distribution de l'eau dans les pays en développement

Les fuites de canalisation d'eau sont l'un problèmes auxquels sont confrontées les entreprises de services publics de l'eau dans le monde entier, car la perte d'eau par les fuites est reconnue comme un problème coûteux dans le monde entier, en raison du gaspillage du précieux liquide, ainsi que du point de vue économique [25]–[27]. Selon une publication de l'UNICEF en 2015 [28], les données ont révélé que plus de 30% des pays en développement et moins développés n'ont pas accès à des sources d'eau potable de qualité. En outre, un autre rapport publié par la Banque mondiale en 2016 [29], a indiqué que dans les pays en développement, environ 45 millions de mètres cubes d'eau sont perdus chaque jour avec une valeur économique de plus de 3 milliards de dollars US par an. Ce rapport indiquait également que l'économie de la moitié de ces pertes permettrait de fournir suffisamment d'eau pour desservir au moins 90 millions de personnes. Au Cameroun, pays en voie de développement subsaharien, le niveau d'eau non rémunérée (NRW), qui est la partie de la quantité totale d'eau produite pour laquelle la compagnie de distribution d'eau ne génère aucun revenu (parce qu'elle est perdue par des fuites/éclatements et/ou des vols) est de 4,67% [30], [31]. Des études récentes ont également révélé que le volume mondial de NRW est de

346 millions de mètres cubes par jour [32]. La raison de ce niveau de NRW est principalement due à l'infrastructure vieillissante du RDE qui crée des pertes physiques par des fuites et/ou des éclatements.

1.2.3 Le besoin de systèmes fiables de détection des fuites en temps réel

La demande en eau augmente continuellement et rapidement en raison de la croissance de la population de la Terre, mais les ressources en eau sont confrontées à une diminution constante causée par le réchauffement de la planète et le changement climatique [25], [33]. Contrairement à d'autres phénomènes plus particuliers, la pénurie d'eau est commune aux pays en développement et aux pays développés [25]. La rareté de l'eau exige donc que les pertes d'eau résultant de fuites soient minimisées en détectant et en localisant précisément les fuites en temps réel, chaque fois qu'elles se produisent. Aussi, étant donné que les fuites exposent l'eau traitée à l'environnement extérieur, il est nécessaire de protéger l'eau potable traitée de la contamination (qui peut l'amener à servir d'avenue potentielle pour l'apparition de maladies) [23], en identifiant rapidement les fuites dans le RDE lorsqu'elles se produisent. Au fil des ans, cela a donné lieu à de nombreuses recherches sur le terrain [34]–[37], [37]–[47], [197], fournissant un large éventail de méthodes pour détecter et localiser les fuites dans les canalisations d'eau.

1.2.4 Surveillance des canalisations d'eau par WSN

Les WSN pour la surveillance des canalisations d'eau - WSN-based Water Pipeline Monitoring (WWPM) - se composent d'un certain nombre de nœuds de capteurs dotés de capteurs à faible coût qui recueillent périodiquement des signaux de fuite sur la canalisation. Les signaux sont ensuite traités pour détecter la présence ou l'absence d'une fuite sur la canalisation. Étant donné que les RDE sont des structures linéaires, le déploiement d'une solution WWPM en tant que WSN linéaire est tout à fait envisageable. Cependant, la plupart des solutions WWPM dans la littérature ont négligé la conception linéaire des WSN [10]. Même si très peu d'études WWPM ont déployé une topologie linéaire [10], nous nous concentrerons dans cette étude sur le déploiement de systèmes WWPM en tant que WSN linéaire.

Il existe plusieurs facteurs affectant les schémas de surveillance des pipelines, tels que le mécanisme de communication, les méthodes d'évaluation, la gestion de l'énergie, les types de surveillance, la connectivité des capteurs, la couverture de détection, les méthodes de détection et les types de capteurs [10]. Nonobstant, le plus grand défi de la détection des fuites dans les systèmes WWPM utilisant des capteurs à faible coût est que les signaux de fuite peuvent être imprécis en raison de la faible sensibilité des capteurs et du bruit environnemental, et peuvent entraîner de fausses alarmes dans le système de détection des fuites. Ainsi, la question de l'identification fiable d'un signal de fuite au milieu d'erreurs provenant d'un certain nombre de sources (communément appelé bruit) est un défi fondamental de tout système de détection de fuites [46], [48].

En fonction de l'endroit où sont traités les signaux de fuite provenant des capteurs distants, les solutions WWPM peuvent être classées comme centralisées, décentralisées ou distribuées. Plusieurs solutions WWPM centralisées ont été proposées dans la littérature [41], [42], [46], [49], [50], où les nœuds de capteurs collectent périodiquement les signaux de fuite de la conduite où ils sont installés et les transmettent à une station de base centrale (où se trouve l'algorithme de détection des fuites) pour un traitement ultérieur afin de détecter la présence d'une fuite sur la conduite. Ces systèmes centralisés se caractérisent par un grand nombre de transmissions multi-sauts et peuvent épuiser plus rapidement l'énergie des nœuds de capteurs. Aussi, certaines études [13], [38], [40], [51] ont proposé des solutions décentralisées où certains traitements pour la détection de fuites sont effectués au niveau du nœud de capteur. Cependant, à notre connaissance, aucune étude n'a proposé une solution entièrement distribuée où tous les traitements nécessaires à la détection des fuites sont effectués par le nœud de capteurs.

1.3 Problématique

1.3.1 Contexte

Depuis plus d'une décennie, les systèmes WWPM sont couramment utilisés pour la surveillance des réseaux de distribution d'eau. Cependant, les solutions WWPM existantes sont confrontées à de nombreux défis tels qu'une faible efficacité énergétique et la difficulté de détecter les fuites en temps réel, car elles sont centralisées et aussi parce qu'elles utilisent des capteurs intrusifs (pression

et débit) qui sont coûteux, difficiles à installer et consomment plus d'énergie. Ces dernières années, les systèmes WWPM utilisant des capteurs de vibration sont devenus populaires, car ces capteurs offrent l'avantage distinct de fournir une surveillance en temps réel du RDE, ce qui peut entraîner des interventions immédiates [35], [43], [46]. Les capteurs de vibrations peuvent être utilisés pour la surveillance, parce que la surveillance de la pression des canalisations d'eau peut être transformée en surveillance des vibrations de la surface de la canalisation [37], [52], [53], puisque un changement transitoire de la pression est toujours accompagné d'une augmentation de l'accélération de la surface de la canalisation aux endroits correspondant sur la longueur de la canalisation [53]. Les fluctuations de pression dans une conduite sont donc liées à la vibration de la surface de la conduite via une relation non linéaire mais proportionnelle [37], [53]. De plus, les capteurs de vibrations (accéléromètres, transducteurs piézoélectriques, résistances sensibles à la force, etc.) sont faciles à installer, moins coûteux à entretenir/exploiter et consomment moins d'énergie.

1.3.2 L'enjeux

Dans la plupart des pays en développement, le réseau de distribution d'eau (RDE) est constitué de tuyaux en plastique et il a été démontré dans la littérature que la propagation des signaux de fuite (vibrations) ne va pas loin dans les tuyaux en plastique [54], [55]. De plus, une détection fiable des fuites nécessitera de placer les capteurs très près les uns des autres pour avoir une résolution spatiale plus élevée [56]. Des accéléromètres de haute précision fixés à la surface extérieure de la conduite peuvent être utilisés pour détecter avec précision cette augmentation soudaine de l'accélération de la surface de la conduite causée par des fuites sur le pipeline. Cependant, la nécessité de distances inter-capteurs plus faibles et la nature onéreuse de ces accéléromètres augmenteront la valeur économique globale et le coût d'installation, ce qui les rend peu adaptés à l'installation dans les pays en développement. Par conséquent, les accéléromètres MEMS à faible coût peuvent constituer une solution réalisable et économiquement viable pour le déploiement dans les pays en développement. Bien que les études précédentes sur les WWPM utilisant des accéléromètres MEMS à faible coût aient été utiles pour détecter les fuites, elles ont toujours le défi de détecter de manière fiable les fuites au milieu de bruits environnementaux aléatoires en raison de la faible précision des capteurs [35], [46], [57], [58]. Un autre défi est l'incapacité de ces

systèmes WWPM précédents à fournir une détection fiable des fuites en temps réel tout en préservant la durée de vie du WSN puisque la plupart d'entre eux sont centralisés. Cependant, étant donné que le WWPM utilise plusieurs capteurs à faible coût pour surveiller le pipeline, les techniques de fusion de données multicapteurs, qui ont été utilisées avec succès dans les applications de suivi de cibles [59], peuvent être utilisées pour augmenter la fiabilité des systèmes de détection de fuites basés sur des capteurs de vibrations à faible coût [57].

1.3.3 Solution proposée et ses avantages

La fusion de données multicapteurs peut combiner les données redondantes de plusieurs capteurs à faible coût pour obtenir une information plus précise dont la qualité dépasse celle obtenue en utilisant un seul capteur [57], [60]. En outre, l'exigence de faible consommation d'énergie et la nécessité pour un système WWPM de rester sans surveillance pendant une longue période sans remplacement de la batterie du nœud de capteur [46], [61], affectent le choix d'une technique de fusion de données multi-capteurs qui peut être utilisée. C'est pourquoi la fusion de données multicapteurs dans les WSN peut être effectuée de manière centralisée, décentralisée ou distribuée [59], [63]. La technique de fusion de données centralisée nécessitera des communications multi-sauts, qui ont une probabilité plus élevée de développer un trou énergétique dans le réseau, réduisant ainsi la durée de vie du WSN [18], [21], [22]. Cependant, l'utilisation de la fusion de données distribuée peut augmenter la durée de vie du WSN car il n'y a pas de point central pour la fusion, et les communications multi-sauts seront entièrement éliminées. L'objectif de la fusion de données distribuée est d'utiliser des calculs distribués à travers le réseau de sorte que les informations locales de chaque nœud de capteur convergent vers une valeur optimale [64].

La mise en œuvre de la fusion de données distribuées dans le WWPM apportera les avantages suivants :

- Elle permettra la réalisation d'une surveillance en temps réel, permettant de détecter les fuites dès qu'elles se produisent. Cela conduira à une intervention rapide, réduisant ainsi la quantité d'eau traitée qui sera perdue et empêchant également la contamination de l'eau potable traitée ;

- Elle a la capacité de réduire la consommation d'énergie des nœuds de capteurs tout en préservant la durée de vie du WSN, ce qui permet au système WWPM d'être opérationnel pendant une longue période sans avoir à remplacer les batteries. En effet, le traitement est effectué au niveau du nœud et le nombre de transmissions multi-sauts est réduit. Cela réduit l'énergie dépensée en communication, qui consomme la majeure partie de l'énergie du nœud de capteur ;
- Le besoin en bande passante du WSN sera réduit puisque le traitement est effectué par l'unité de traitement du nœud de capteur et que les nœuds de capteur ne communiqueront avec les nœuds voisins que pour obtenir une plus grande précision. Ceci est avantageux car les WSN sont limités en bande passante.

Inspirés par l'application d'accéléromètres MEMS à faible coût dans les systèmes WWPM et par les défis de ces systèmes (tels que la surveillance en temps réel, les exigences de faible coût et de faible consommation d'énergie, etc.) qui n'ont pas été correctement traités dans les études précédentes. Nous avons décidé d'étudier l'application du filtre de Kalman distribué (DKF), qui est un filtre de Kalman avec fusion de données distribuées, comme une solution entièrement distribuée (calcul distribuée) pour la détection de fuites dans les systèmes WWPM utilisant des accéléromètres MEMS à faible coût. C'est pourquoi la proposition d'une solution de calcul distribué qui est en temps réel, à faible puissance, et qui détecte de manière fiable les fuites dans les systèmes WWPM utilisant des accéléromètres MEMS à faible coût sera le point central de cette thèse.

1.4 Objectifs

L'objectif principal de cette thèse est d'établir une augmentation de la performance et de la durée de vie d'un WSN via l'utilisation d'un paradigme de calcul distribué. Le WWPM étant un domaine d'application du WSN qui nécessite une faible consommation d'énergie [61], le défi est d'atteindre à la fois une haute précision de détection des fuites et une durée de vie maximale du système WWPM. De plus, étant donné que les pertes d'eau dues aux fuites sont un défi majeur auquel sont confrontées les compagnies de distribution d'eau dans la plupart des pays en voie de développement et aussi parce que le système WWPM présente un bon scénario d'application où le calcul distribué efficace peut être démontré dans les WSN (en

raison de leur topologie linéaire 1D qui les rend plus faciles à gérer pour une première démonstration), nous concentrerons notre étude sur l'examen de la possibilité de déployer une solution en temps réel, fiable, à faible coût, à faible puissance et entièrement distribuée pour le système WWPM en utilisant des capteurs de vibrations à faible coût et DKF. Aussi, le but de cette étude est d'examiner l'application de DKF dans la réalisation d'une détection de fuites fiable, en temps réel et à faible consommation d'énergie dans le WWPM en utilisant les données de vibration des accéléromètres MEMS à faible coût.

Le but de cette recherche sera atteint en répondant aux objectifs spécifiques suivants :

1. Passer en revue les techniques populaires de surveillance des canalisations d'eau disponibles dans la littérature en mettant l'accent sur les solutions WWPM qui utilisent des capteurs de vibration à faible coût pour la surveillance. A partir de là, identifier les lacunes dans les connaissances dans le domaine de la surveillance des conduites en plastique à l'aide d'accéléromètres MEMS à faible coût ;
2. Établir les spécifications d'un nœud de capteurs à faible coût et à faible puissance capable d'effectuer un traitement *in situ* sous contraintes énergétiques. Le nœud de capteurs doit avoir une capacité de calcul suffisante pour effectuer un traitement local et être sensible à la consommation d'énergie. Sur cette base, concevoir et mettre en œuvre un nœud de capteurs sans fil fondé sur les spécifications établies ;
3. Proposer une solution WWPM entièrement distribuée, en temps réel et à faible consommation d'énergie, basée sur DKF pour une détection fiable des fuites dans les tuyaux en plastique. À cet égard, nous effectuerons une recherche bibliographique approfondie sur les algorithmes DKF pouvant être mis en œuvre dans les WSN et sélectionnerons trois algorithmes DKF dont les performances de détection de fuites et la consommation d'énergie seront évaluées dans le contexte de la WWPM à l'aide d'accéléromètres MEMS à faible coût ;
4. Démontrer la faisabilité de l'application d'un DKF pour une détection fiable des fuites en temps réel dans un système WWPM qui utilise des accéléromètres MEMS à faible coût en déployant un réseau de capteurs qui met en œuvre l'un des algorithmes DKF sélectionnés. Sur cette base, évaluer les performances de détection de fuites et la

consommation d'énergie de la solution distribuée proposée en effectuant des simulations et en validant les résultats de simulation à l'aide des résultats obtenus lors d'expériences physiques sur un banc d'essai de laboratoire ;

5. Évaluer les performances de détection des fuites et la consommation d'énergie des algorithmes DKF sélectionnés à l'aide de simulations et d'expériences physiques. Sur cette base, comparer les performances de détection de fuites et la consommation d'énergie des algorithmes DKF sélectionnés et proposer lequel des DKF est optimal pour la mise en œuvre dans les systèmes WWPM utilisant des accéléromètres MEMS à faible coût.

1.5 Contributions

Cette thèse se concentre principalement sur la proposition d'une solution entièrement distribuée et à faible consommation d'énergie pour une détection fiable et en temps réel des fuites dans les systèmes WWPM en utilisant des accéléromètres MEMS à faible coût. Bien que des recherches approfondies aient déjà été menées sur les systèmes WWPM utilisant des capteurs de vibrations à faible coût, plusieurs défis subsistent, tels que la fourniture d'un système de détection de fuites entièrement distribué, en temps réel, à faible coût et consommant moins d'énergie. En outre, la mise en œuvre et l'évaluation d'algorithmes de détection de fuites entièrement distribués applicables aux WSN du monde réel et qui utilisent des approches distribuées sont rares [10]. Sur la base des lacunes dans les connaissances de la littérature, ce travail est nouveau en ce sens qu'il est le premier à appliquer DKF dans le contexte de la détection de fuites dans les systèmes WWPM utilisant des accéléromètres MEMS à faible coût.

Cette thèse a apporté des contributions à la recherche qui améliorent non seulement l'état de l'art de la détection de fuites en temps réel dans les systèmes WWPM en utilisant des accéléromètres MEMS à faible coût, mais aussi celui des applications WSN linéaires. Ces contributions comprennent :

- La proposition d'un nœud de capteurs personnalisé à faible coût basé sur des composants commerciaux bon marché (COTS) tels que le microcontrôleur ESP32 (unité de traitement), l'émetteur-récepteur nRF24L01+ (unité de communication) et les accéléromètres ADXL344 et LSM9DS1 (unité de détection) ;

- Le développement d'un nœud de capteurs personnalisés à faible consommation mais à forte capacité de calcul, basé sur l'ESP32 et son coprocesseur Ultra-low Power (ULP), l'émetteur-récepteur nRF24L01+ avec sa faible consommation et son mode rafale, et la capacité de détection de seuil de l'ADXL344 et du LSM9DS1 ;
- La démonstration d'une solution de détection des fuites entièrement distribuée, fiable et en temps réel, en mettant en œuvre un algorithme DKF pour améliorer la précision de l'accélération de la surface des tuyaux mesurée par les nœuds de capteurs dans un système WWPM ;
- L'évaluation de la performance et de la consommation d'énergie de la solution de détection de fuites proposée basée sur le DKF via des simulations dans CupCarbon 4.2 et la validation sur un banc d'essai expérimental. La comparaison des résultats de simulation et des résultats expérimentaux est une contribution essentielle à la mise en œuvre réelle de l'approche proposée, car de nombreuses études dans la littérature ne vont pas jusqu'à l'analyse expérimentale et l'étude du comportement des nœuds de capteurs dans des conditions réelles. Les résultats ont montré que la capacité de fusion de données distribuées du DKF améliore la fiabilité de la détection des fuites tout en préservant la durée de vie du WSN ;
- L'évaluation de la performance et de la consommation d'énergie de trois algorithmes DKF sélectionnés via la simulation et la validation sur un banc d'essai expérimental. Les résultats ont montré que les algorithmes DKF basés sur la diffusion consomment moins d'énergie (préservent la durée de vie du WSN) mais ont des performances inférieures en matière de détection de fuites, tandis que les algorithmes DKF basés sur le consensus, qui consomment plus d'énergie en raison du nombre accru de communications, offrent une meilleure fiabilité en matière de détection de fuites ;
- La mise en œuvre d'un cycle de travail et d'une détection hiérarchique au niveau des nœuds de capteurs pour réduire la consommation d'énergie du système WWPM entièrement distribué proposé.

Dans l'ensemble, l'étude et l'évaluation d'une solution peu coûteuse, de faible puissance et entièrement distribuée basée sur DKF pour la détection de fuites en temps réel dans les systèmes

WWPM en utilisant des accéléromètres MEMS peu coûteux est la contribution originale à la connaissance de cette thèse.

1.6 Portée de la thèse

Étant donné que le filtre de Kalman peut être appliqué en tant qu'observateur d'état et algorithme de fusion de données, nous nous concentrons dans cette étude sur l'utilisation du filtre de Kalman en tant qu'algorithme de fusion de capteurs pour améliorer la précision des mesures de fuites collectées par les nœuds de capteurs locaux. En tant que technique d'observation de l'état, le filtre de Kalman est utilisé pour calculer une estimation optimale d'un paramètre intéressant (par exemple, la taille et l'emplacement de la fuite) à partir d'informations disponibles limitées (par exemple, la pression et le débit), tandis qu'en tant qu'algorithme de fusion de capteurs, le filtre de Kalman est utilisé pour l'estimation optimale de l'état d'un paramètre basé sur des données provenant de modèles dynamiques et de mesures de capteurs ou la fusion de mesures provenant de plusieurs capteurs. Les méthodes de filtre de Kalman utilisées pour la détection des fuites dans les réseaux de distribution d'eau ont été largement appliquées aux mesures de pression et de débit. Dans la littérature, le filtre de Kalman et ses variantes ont été utilisés comme des techniques basées sur des modèles pour la détection de fuites avec des mesures de débit et de pression provenant de capteurs intrusifs servant de sources de données. Cependant, dans cette thèse, nous utilisons le filtre de Kalman comme un algorithme de traitement du signal agissant sur les mesures des accéléromètres MEMS et effectuant la fusion des données afin d'isoler les signaux de fuite du bruit. En outre, dans un premier temps, nous étudions uniquement les algorithmes DKF qui peuvent être mis en œuvre sur des nœuds de capteurs sans fil.

1.7 Présentation de la thèse

Cette thèse contient 7 chapitres et est organisée comme suit :

- Le chapitre 2 présente un historique des techniques de gestion de l'énergie et du calcul distribué dans les WSN. Il fournit également une analyse documentaire complète sur les solutions de gestion de l'énergie dans les WSN et les lacunes en matière de connaissances ;

- Le chapitre 3 présente la sélection des composants COTS utilisés pour la conception et la mise en œuvre d'un nœud de capteurs personnalisé qui sera déployé sur le banc d'essai du laboratoire ;
- Le chapitre 4 présente une étude des algorithmes DKF pour la mise en œuvre dans les WSN et la sélection de trois DKF qui seront mis en œuvre et dont les performances de détection de fuites et la consommation d'énergie seront évaluées dans le contexte du WWPM en utilisant des accéléromètres MEMS à faible coût ;
- Le chapitre 5 présente les résultats d'une première démonstration de l'application de DKF dans la réalisation d'une solution de détection de fuites entièrement distribuée, en temps réel, fiable et de faible puissance pour les systèmes WWPM utilisant des accéléromètres MEMS à faible coût ;
- Le chapitre 6 présente les résultats de l'évaluation des performances de détection de fuites et de la consommation d'énergie des trois DKF sélectionnés pour les expériences menées sur une plateforme de simulation et un banc d'essai de laboratoire. Il présente également les résultats obtenus par la mise en œuvre expérimentale du cycle de fonctionnement et de la détection hiérarchique sur les nœuds de capteurs ;
- Le chapitre 7 présente un certain nombre de conclusions tirées de la réalisation des objectifs de cette étude et fournit également des recommandations pour les travaux futurs.

Chapitre 2 : Calcul distribué et surveillance des conduites d'eau à base de capteurs sans fil

2.1 Introduction

Ce chapitre présente le contexte et la revue de la littérature de notre étude. Il commence par classer les applications de surveillance WSN en trois catégories : centralisées, décentralisées et distribuées, en fonction du lieu et de la manière dont les données collectées au sein du WSN sont traitées. Cette classification est suivie d'une discussion sur les défis des applications centralisées de surveillance WSN, qui sont prévalents dans la littérature. Les techniques de gestion de l'énergie disponibles dans la littérature pour maximiser la durée de vie des WSN sont passées en revue. Les techniques de gestion de l'énergie sont classées en trois catégories : récolte d'énergie, équilibrage de l'énergie et conservation de l'énergie. Ensuite, une discussion sur le calcul distribué dans les WSN et ses avantages est présentée. Un certain nombre d'implémentations de le calcul distribuée dans les WSN, classées dans les catégories suivantes : traitement distribué des requêtes, traitement collaboratif des signaux, estimation/détection distribuée de l'état et traitement au sein du réseau, sont par la suite discutées. Nous présentons ensuite la surveillance des canalisations d'eau en mettant l'accent sur les systèmes WWPM utilisant des capteurs de vibration à faible coût. Nous mentionnons les défis posés par les méthodes de détection des fuites basées sur les vibrations dans les conduites en plastique et la nécessité d'améliorer la fiabilité de la détection des fuites. Nous passons également en revue les solutions WWPM disponibles dans la littérature et les classons comme centralisées, décentralisées et distribuées en fonction de l'endroit où sont exécutés les algorithmes de prétraitement, de détection et de localisation des fuites. Enfin, nous examinons l'application des techniques d'économie d'énergie pour prolonger la durée de vie des systèmes WWPM et nous présentons un résumé des lacunes identifiées dans les connaissances.

Ce chapitre est organisé comme suit :

La section 2.1 présente la classification des applications WSN en fonction de l'architecture du réseau et de la stratégie de traitement des données employée. La classification des WSN en fonction du traitement des données est déterminée par l'endroit où le traitement des données est

mis en œuvre (nœud de capteurs ou station de base), c'est-à-dire si la station de base reçoit des données brutes, des données traitées ou des décisions des nœuds de capteurs [65]. En outre, elle affecte les coûts de calcul et de communication et influence également l'évolutivité du réseau et le type d'applications WSN où elle peut être appliquée [63]. En général, selon la technique de traitement des données, les applications de surveillance WSN peuvent être classées en trois catégories [40], [59], [60], [63], [65]: surveillance centralisée, surveillance décentralisée ou surveillance distribuée. Dans les WSN centralisés, tous les nœuds de capteurs envoient leurs données brutes, via des transmissions à un ou plusieurs sauts, à un seul centre de fusion (station de base ou puits) qui traite ensuite les données brutes. Les WSN distribués ne nécessitent pas de centre de fusion et tout le traitement des données est effectué au sein du nœud de capteur. Les WSN décentralisés sont un compromis entre les WSN complètement centralisés et les WSN distribués. Ils sont constitués de plusieurs centres de fusion qui reçoivent les données brutes ou partiellement traitées des nœuds de capteurs, et sont capables de communiquer avec leurs proches voisins ou directement avec la station de base.

La section 2.2 présente les inconvénients des applications centralisées de surveillance WSN. Ces inconvénients sont les suivants : manque d'évolutivité, faible efficacité énergétique, augmentation de la latence, réduction de la robustesse et de la fiabilité, et faible confidentialité et sécurité des données. Cette section montre que l'efficacité énergétique est une exigence principale dans la conception efficace des WSN qui doivent être opérationnels pendant une longue période sans remplacement des sources d'énergie des nœuds de capteurs.

La section 2.3 aborde brièvement les techniques de prolongation de la durée de vie des WSN disponibles dans la littérature. Les nœuds de capteurs étant généralement des dispositifs alimentés par batterie, la durée de vie du WSN est limitée par la batterie des capteurs individuels du WSN [16]. Ainsi, les questions critiques à prendre en compte pour maximiser la durée de vie du WSN sont de savoir comment réduire la consommation d'énergie des nœuds ou comment reconstituer leurs sources d'énergie de manière efficace et réaliste [16], [71] ? Il est nécessaire que la durée de vie du WSN soit suffisamment longue pour permettre au WSN de répondre aux exigences de l'application. Ces dernières années, cela a conduit à une pléthore de techniques de gestion de l'énergie dans la littérature pour prolonger la durée de vie du WSN. Les techniques de prolongation

de la durée de vie des WSN sont classées en trois grandes catégories : les techniques de conservation de l'énergie, les techniques d'équilibrage de l'énergie et les techniques de récolte de l'énergie, chacune d'entre elles comportant un certain nombre de sous-catégories. Dans cette section, on voit que le calcul distribué a le potentiel de réduire la consommation d'énergie des nœuds de capteurs et, par conséquent, de prolonger la durée de vie du WSN, car il réduit le nombre de messages et la quantité de données transmises en utilisant les ressources informatiques embarquées du nœud de capteur.

La section 2.4 présente une brève revue du calcul distribué dans les WSN en examinant la motivation du calcul distribué dans les WSN, les avantages du calcul distribué dans les WSN et en présentant un bref aperçu de certaines études qui ont appliqué le calcul distribué dans les WSN.

La section 2.5 présente une classification générale des techniques de détection des fuites et une taxonomie que nous avons développée pour classer les solutions de détection des fuites. Selon cette taxonomie, les techniques de détection des fuites utilisées pour la WPM peuvent être classées comme statiques ou dynamiques, en fonction de la mobilité de l'équipement utilisé pour recueillir les signaux de fuite. Selon que la technique de détection des fuites surveille des paramètres externes ou internes de la conduite d'eau et selon l'utilisation d'un équipement matériel spécialisé ou de capteurs associés à des algorithmes de calcul pour traiter les signaux de fuite, les techniques de détection des fuites sont classées en méthodes externes (matérielles) ou internes (logicielles). Les méthodes basées sur des logiciels sont également classées en sous-catégories qui utilisent les WSN pour surveiller le pipeline (appelées techniques WWPM) et en sous-catégories qui utilisent d'autres technologies que les WSN (par exemple, la télémétrie câblée et le contrôle et l'acquisition de données (Supervisory Control and Data Acquisition -SCADA), qui utilisent des capteurs connectés au centre de contrôle principal par des moyens de communication tels que des câbles en cuivre ou des fibres optiques), auquel cas elles sont appelées techniques non basées sur les WSN. Les techniques WWPM sont ensuite classées en deux catégories : les méthodes intrusives qui utilisent des capteurs invasifs pour surveiller les paramètres internes du pipeline, tels que la pression et/ou le débit, et les méthodes non intrusives qui utilisent des capteurs non invasifs, tels que des accéléromètres et/ou des capteurs acoustiques, pour surveiller les paramètres externes du pipeline, tels que l'accélération de la surface de la conduite, les signaux sonores de fuite, etc. Enfin,

en fonction de la technique d'analyse des signaux ou des données de fuite, les techniques de détection des fuites sont classées en méthodes basées sur des modèles, sur des transitoires, sur le traitement des signaux ou sur les données.

La section 2.6 présente la surveillance des canalisations d'eau (WWPM) basée sur le WSN. La sous-section 2.6.2 examine en détail les défis posés par l'utilisation de capteurs non intrusifs pour la détection de fuites dans les conduites en plastique, notamment la courte distance entre les capteurs et le taux élevé de fausses alarmes. La sous-section 2.6.3 présente certains algorithmes de calcul basés sur des techniques de traitement du signal qui ont été utilisés dans des systèmes WWPM utilisant des capteurs non intrusifs pour améliorer la précision de la détection des fuites. La sous-section 2.6.4 présente un état des lieux complet de la WWPM avec une analyse et une comparaison détaillée des études importantes dans le domaine qui utilisent des capteurs non intrusifs. Il ressort de cet examen que les accéléromètres MEMS à faible coût constituent une solution peu coûteuse et de faible puissance pour la détection des fuites dans les RDE en plastique. Enfin, la sous-section 2.6.5 passe en revue les études qui ont appliqué une ou plusieurs techniques de gestion de l'énergie pour prolonger la durée de vie du réseau (présentée à la section 2.3) dans le contexte du WWPM.

2.2 Synthèse et lacunes identifiées dans les connaissances

2.2.1 Synthèse

Obtenir une détection précise des fuites en temps réel tout en préservant la durée de vie du système WWPM pendant une longue période est un défi majeur dans WWPM. Comme cela a été montré dans la littérature, la plupart des systèmes WWPM disponibles sont centralisés et décentralisés. Les systèmes centralisés ont tendance à consommer beaucoup d'énergie et ne permettent pas de surveillance en temps réel, ce qui en fait pas une solution optimale pour la détection des fuites dans WWPM. Au contraire, très peu d'études WWPM fournissent une solution entièrement distribuée pour la détection des fuites. Cependant, ces dernières années, il y a eu une augmentation du nombre d'études prônant le développement de solutions distribuées [10], [34].

De plus, la question de la conservation de l'énergie est d'une importance capitale dans le cadre de la WWPM. Cependant, la plupart des études WWPM se concentrent sur la localisation des fuites tout en négligeant la consommation d'énergie de leurs solutions WWPM proposées. Ainsi, il existe très peu d'études dans la littérature qui s'intéressent à l'évaluation de la consommation d'énergie de leur solution WWPM proposée et aux moyens de conserver l'énergie afin de prolonger la durée de vie du système WWPM. Par ailleurs, la plupart des études qui ont traité de la consommation d'énergie de leurs solutions WWPM proposées se sont basées sur des simulations et très peu d'études vont jusqu'à réaliser des expériences physiques sur un banc d'essai de laboratoire ou un RDE réel.

Récemment, les études visant à obtenir une détection fiable des fuites dans les canalisations en plastique à l'aide de capteurs non intrusifs comme les accéléromètres sont devenues populaires, faisant de la WWPM utilisant des accéléromètres un domaine de recherche très actif. Dans la littérature, la plupart des études ont proposé l'utilisation d'accéléromètres de haute précision et de puissants algorithmes de traitement du signal pour une détection fiable des fuites. Cependant, ces solutions ne sont pas idéales pour les exigences de faible coût et de faible puissance des systèmes WWPM car elles négligent à la fois le coût et l'efficacité énergétique. Même si un certain nombre d'études récentes ont proposé l'utilisation d'accéléromètres MEMS à faible coût pour la détection de fuites dans les tuyaux en plastique. Le problème combiné de la précision de la détection des fuites et de la consommation d'énergie n'a pas été bien traité. À notre connaissance, aucune étude n'a jusqu'à présent examiné ou appliqué des algorithmes de fusion de données distribuées redondantes tels que DKF (qui ont été largement utilisés pour augmenter la précision des systèmes de suivi) dans le domaine du WWPM.

De plus, conscients du fait que le choix des composants qui constituent un capteur affecte le coût, la consommation d'énergie et les performances globales du nœud, nous avons remarqué dans la littérature que peu d'études sur les systèmes WWPM s'intéressaient réellement au choix des composants qui constituent leurs nœuds de capteurs. Afin d'atteindre les objectifs de faible coût, de faible consommation, de temps réel et de haute fiabilité (qui sont des exigences contradictoires) dans les systèmes WWPM, il est nécessaire de bien examiner la sélection des éléments COTS utilisés comme blocs de construction des nœuds de capteurs.

Aussi, pour fournir une solution WWPM fiable, en temps réel, entièrement distribuée et à faible consommation d'énergie pour la surveillance des tuyaux en plastique, trois éléments seront nécessaires :

1. Un nœud de capteurs doté de ressources informatiques suffisantes pour effectuer un traitement *in situ* tout en maintenant une faible consommation d'énergie grâce à une mise en œuvre efficace du cycle de fonctionnement ;
2. Des capteurs pour réduire l'énergie de détection via la mise en œuvre d'une détection hiérarchique ;
3. Une technique de traitement du signal qui nécessite moins de calculs et qui a la capacité de prédire les données détectées afin de réduire les coûts de communication et la capacité d'effectuer une fusion de données distribuées afin d'augmenter la précision de la détection des fuites.

2.2.2 Lacunes identifiées dans les connaissances

Suite à la recherche documentaire qui a été menée, les lacunes identifiées dans les connaissances sont résumées ci-dessous :

1. Pendant que plusieurs études dans la littérature sont impliquées dans la détection de fuites dans les systèmes WWPM, la plupart des solutions utilisent soit un traitement centralisé ou décentralisé des données pour détecter les fuites. À notre connaissance, peu de travaux ont été réalisés pour mettre en œuvre une solution en temps réel et entièrement distribuée pour la détection de fuites dans les tuyaux en plastique en utilisant un accéléromètre MEMS à faible coût ;
2. Alors que la plupart des études WWPM se concentrent sur la détection et la localisation des fuites, très peu d'études ont évalué la consommation d'énergie de leur solution et ont cherché des moyens de réduire la consommation d'énergie afin de prolonger la durée de vie de la surveillance ;
3. Aucune étude n'a combiné la mise en œuvre de techniques de conservation de l'énergie par cycle de service, de détection hiérarchique et de prédiction des données au niveau des

nœuds de capteurs pour prolonger la durée de vie des systèmes WWPM utilisés pour la surveillance des RDE en plastique ;

4. Aucune étude n'a examiné une solution WWPM entièrement distribuée basée sur DKF pour améliorer la précision de la détection des fuites dans les tuyaux en plastique surveillés par des accéléromètres MEMS à faible coût, ni évalué sa consommation d'énergie ;
5. Alors que plusieurs études ont évalué les performances de différents algorithmes DKF dans des applications de suivi de cible, aucune étude n'a évalué à la fois les performances de détection de fuite et la consommation d'énergie des algorithmes DKF via des simulations et des expériences physiques sur un banc d'essai de laboratoire dans le contexte du WWPM des RDE en plastique.

Afin de combler les lacunes dans les connaissances identifiées ci-dessus, nous proposons de répondre aux questions de recherche suivantes :

1. Quels composants COTS seront utilisés pour répondre aux exigences d'un nœud de capteur à faible coût et à faible puissance qui dispose de ressources informatiques suffisantes pour effectuer un traitement *in situ* ?
2. Comment le nœud de capteurs sera-t-il conçu, mis en œuvre et configuré de manière à obtenir une capacité de calcul élevée et une faible consommation d'énergie ?
3. Quelles techniques de gestion de l'énergie seront utilisées pour réduire la consommation d'énergie des nœuds de capteurs et augmenter la durée de vie du WSN ?
4. Quelle technique de traitement du signal sera appropriée (en termes de précision et de consommation d'énergie) pour traiter les signaux de fuite obtenus à partir de tuyaux en plastique en utilisant des accéléromètres MEMS à faible coût ?
5. La fusion distribuée des données améliorera-t-elle la précision de la détection des fuites ?
6. Quels algorithmes de fusion de données distribuées, avec KF, peuvent être mis en œuvre sur les nœuds de capteurs pour fournir une détection de fuites en temps réel et entièrement distribuée dans les réseaux de distribution d'eau en plastique ?
7. Comment les différentes techniques de fusion de données distribuées affectent-elles la précision de la détection des fuites et la consommation d'énergie du système WWPM ?

Les chapitres suivants de cette thèse fourniront des réponses aux questions de recherche soulignées ci-dessus.

Chapitre 3 : Conception des nœuds de capteurs

L'objectif de ce chapitre est de déterminer les spécifications d'un nœud économique et de faible puissance, doté d'une puissance de calcul suffisante (requise pour le traitement au sein du nœud), qui sera construit à partir de composants commerciaux bon marché (COTS). Ceci est important car nous avons l'intention de réaliser une détection de fuites en temps réel en implémentant tout le traitement requis pour la détection de fuites dans le nœud de capteur et sans avoir besoin de communications longue distance via de multiples sauts vers une station de base. Il est donc nécessaire que l'unité de traitement du nœud de capteurs dispose de ressources informatiques suffisantes pour effectuer les calculs requis et consomme moins d'énergie puisque les nœuds sont censés fonctionner sur batterie. En outre, étant donné que de nombreux nœuds de capteurs seront nécessaires pour surveiller les réseaux de distribution d'eau en plastique (car les signaux de vibration ne vont pas loin dans les tuyaux en plastique), il est nécessaire que le nœud de capteur soit peu coûteux.

Dans ce chapitre, nous présentons la conception et la configuration matérielle d'un nœud de capteurs personnalisé et d'un dispositif de mesure de la puissance que nous utiliserons pour évaluer la solution WWPM entièrement distribuée que nous proposons sur un dispositif expérimental. Tout d'abord, nous trouvons une réponse à la question de savoir quels composants COTS seront optimaux pour réaliser un nœud de capteurs à faible coût et à faible puissance avec une capacité de calcul suffisante pour le traitement dans le nœud. Par la suite, nous cherchons une réponse à la question suivante : quelles sont les spécifications d'un nœud de capteur adapté pour fournir une solution WWPM entièrement distribuée, en temps réel et à faible consommation d'énergie pour surveiller les RDE en plastique? Pour répondre à ces questions, nous commençons ce chapitre par un aperçu général des éléments constitutifs d'un système WWPM, puis nous passons en revue les plates-formes matérielles WSN existantes (commerciales et de recherche) en nous concentrant sur les unités de traitement et de communication. Nous passons ensuite en revue les accéléromètres MEMS à faible coût qui peuvent être utilisés pour mesurer l'accélération de la surface des tuyaux en plastique. Ensuite, nous discutons de la sélection des composants appropriés pour notre nœud de capteurs personnalisé et de la raison de leur choix, en fonction des exigences spécifiques de notre application. Cette discussion est suivie d'une discussion sur la conception et la configuration

du nœud de capteurs. Le chapitre se termine par la présentation de la conception et de la configuration d'un dispositif de mesure de puissance personnalisé pour mesurer la consommation électrique des nœuds de capteurs sur un banc d'essai de laboratoire pendant les expériences physiques.

Ce chapitre est organisé comme suit :

La section 3.1 présente les éléments constitutifs d'un système WWPM, qui se compose en gros d'une couche de capteurs et d'une couche de nuages, puis examine les éléments constitutifs des nœuds de capteurs qui composent la couche de capteurs du système WWPM ;

La section 3.2 passe en revue les unités de traitement MCU et les émetteurs-récepteurs RF des nœuds de capteurs sans fil commerciaux et de recherche. Cet examen a pour but de faciliter la sélection d'une unité de traitement MCU et d'un émetteur-récepteur RF à faible coût et à faible puissance en vue de leur mise en œuvre dans un nœud de capteurs personnalisé qui sera déployé sur le banc d'essai expérimental ;

La section 3.3 présente une revue des accéléromètres MEMS à faible coût et finalement la sélection de deux accéléromètres MEMS (LSMDS1 et ADXL344) ;

La section 3.4 décrit en détail les spécifications des composants COTS utilisés pour concevoir le nœud de capteurs personnalisé. Le nœud de capteurs personnalisé se compose d'un ESP32 d'Espressif Systems comme unité de traitement, d'un module émetteur-récepteur nRF24L01+ de Nordic comme unité de communication et d'un accéléromètre LSM9DS1 de STMicroelectronics et d'un accéléromètre ADXL344 d'Analog Devices comme unité de détection ;

La section 3.5 décrit la configuration des composants matériels du nœud de capteur personnalisé. Le module émetteur-récepteur nRF24L01+ est interfacé avec l'ESP32 via l'interface SPI, tandis que les capteurs LSM9DS1 et ADXL344 sont interfacés avec l'ESP32 via les interfaces I2C.

La section 3.6 traite de la conception et de la configuration d'un dispositif de mesure de l'énergie personnalisé qui sera utilisé pour mesurer la consommation d'énergie des nœuds de capteurs dans le dispositif expérimental.

Chapitre 4 : Filtre de Kalman distribué pour les réseaux de capteurs sans fil

L'objectif de cette étude est d'examiner et d'évaluer la mise en œuvre d'une solution WWPM fiable, en temps réel, entièrement distribuée, à faible coût et à faible puissance pour la détection des fuites dans les réseaux de distribution d'eau en plastique. Nous visons à atteindre :

1. Un bon rapport coût-efficacité en utilisant des composants COTS à faible coût comme éléments constitutifs du nœud de capteur ;
2. Une faible consommation d'énergie en mettant en œuvre un cycle de travail et une détection hiérarchique sur le nœud de capteur ;
3. Une détection des fuites fiable et en temps réel en mettant en œuvre un calcul distribué dans le système WWPM.

Dans le chapitre 3, nous avons discuté des spécifications d'un nœud de capteurs capable d'effectuer des traitements *insitu.* Nous avons également conçu un nœud de capteurs personnalisé à partir de composants COTS à faible coût et proposé une méthode pour réduire la consommation d'énergie du nœud de capteurs en mettant en œuvre un cycle de fonctionnement et une détection hiérarchique. Le contenu du chapitre 3, dans une certaine mesure, réalise les points 1 et 2 énumérés ci-dessus. Dans ce chapitre, l'objectif est de sélectionner des algorithmes de filtre de Kalman distribué (DKF) à faible intensité de calcul à mettre en œuvre dans les nœuds de détection pour permettre une détection fiable des fuites en temps réel tout en préservant la durée de vie du système WWPM. Le défi est qu'il existe de nombreux algorithmes DKF dans la littérature avec des performances, des complexités, des exigences de calcul, etc. différentes. Il est donc nécessaire de sélectionner des algorithmes DKF adaptés aux applications WSN. L'objectif de ce chapitre est de nous rapprocher de la réalisation du point 3 ci-dessus.

Nous commençons ce chapitre par une présentation du Filtre de Kalman (FK) avec les raisons pour lesquelles nous l'avons choisi comme algorithme de traitement du signal pour améliorer la qualité des mesures de vibrations de surface des tuyaux collectées à l'aide d'accéléromètres MEMS à faible coût. Nous passons ensuite en revue les catégories d'algorithmes DKF pour les applications WSN à faible coût telles que les systèmes WWPM. Nous sélectionnons ensuite trois algorithmes DKF

que nous implémenterons et évaluons leurs performances et leur consommation d'énergie via des simulations et des expériences physiques. Nous terminons le chapitre en présentant les mesures permettant d'évaluer les performances et la consommation d'énergie de la solution entièrement distribuée que nous proposons pour la détection des fuites dans les systèmes WWPM.

Ce chapitre est organisé comme suit :

La section 4.1 donne un aperçu de l'algorithme KF standard et présente les techniques de mise en œuvre de la fusion de données au niveau des nœuds de capteurs ;

La section 4.2 aborde les différentes techniques de mise en œuvre de l'algorithme KF dans les WSN. Elle commence par donner les raisons pour lesquelles le KF a été choisi pour être utilisé dans cette étude, puis passe à la discussion du filtre de Kalman centralisé (CKF) et du filtre de Kalman distribué (DKF), en soulignant leurs avantages et inconvénients ;

La section 4.3 présente trois catégories différentes d'algorithmes DKF qui peuvent être mis en œuvre dans les WSN à faible coût et met en évidence leurs avantages et inconvénients ;

La section 4.4 présente la sélection finale de trois algorithmes DKF (un de chaque catégorie présentée dans la section 4.3) et décrit en détail leur fonctionnement dans le contexte du WWPM.

La section 4.5 présente les métriques qui seront utilisées pour évaluer les performances et la consommation d'énergie de notre solution WWPM proposée.

Chapitre 5 : Démonstration d'une solution de détection de fuites basée sur le DKF dans les systèmes WWPM utilisant des accéléromètres MEMS à faible coût

Le premier objectif de ce chapitre est de démontrer le calcul distribué dans un système WWPM composé d'un réseau d'accéléromètres MEMS à faible coût surveillant un RDE en plastique. Etant donné qu'il existe plusieurs applications du calcul distribué dans les WSN, comme mentionné dans la sous-section 2.4.3 du chapitre 2, nous allons cependant démontrer dans ce chapitre le calcul distribué via le traitement local (implémenté par un filtre de Kalman local) et le calcul distribué via l'estimation d'état distribuée (implémenté par un filtre de Kalman distribué). Le deuxième objectif est de montrer que la fusion de données distribuée mise en œuvre par les algorithmes DKF consomme moins d'énergie que la fusion de données centralisée mise en œuvre par CKF et fournit une meilleure précision de détection des fuites par rapport à la mise en œuvre du filtre de Kalman local (LKF) (qui n'implique pas la fusion des estimations locales des nœuds de capteurs voisins). Cet aspect est particulièrement important en raison de la nécessité de détecter de manière fiable les fuites en temps réel, au moment où elles se produisent, afin de réduire les pertes d'eau traitée. De plus, l'obligation pour un système WWPM d'être opérationnel pendant de longues périodes avec une alimentation par batterie nécessite une solution à faible consommation d'énergie afin d'étendre sa durée de vie. En outre, l'utilisation de capteurs à faible coût imposée par l'exigence de faible coût du système WWPM affecte la précision de la détection des fuites. Ainsi, une fusion efficace des données multi-capteurs peut être utilisée comme technique de filtrage pour améliorer la fiabilité du système WWPM tout en préservant sa durée de vie. Enfin, étant donné que les ressources informatiques embarquées des nœuds de capteurs effectueront tout le traitement nécessaire à la détection des fuites, il est impératif que les sorties des nœuds de capteurs voisins soient cohérentes pour éviter les résultats contradictoires de la détection des fuites. Le défi est donc de parvenir à une détection fiable des fuites en temps réel et à une faible consommation d'énergie dans un système WWPM où les nœuds de capteurs sont constitués d'accéléromètres MEMS à faible coût, capables d'effectuer un traitement local et d'échanger des informations avec leurs proches voisins.

Ce chapitre présente les résultats et les discussions de la première série d'expériences qui ont été menées pour notre étude. Il commence par discuter de la composition et de la configuration des nœuds de capteurs et de la mise en œuvre des algorithmes KF (LKF, CKF et DKF). Nous décrivons ensuite la configuration de simulation utilisée pour effectuer les simulations et la configuration de laboratoire utilisée pour valider les résultats obtenus à partir des simulations. Ensuite, nous présentons les résultats et les discussions de la première série d'expériences, où nous avons implémenté un algorithme DKF et un algorithme LKF et comparé leurs performances en utilisant les résultats des simulations et des expériences physiques. Nous évaluons ensuite la consommation d'énergie de l'implémentation DKF par le biais de simulations (en utilisant les valeurs d'un modèle de fiche technique) et sa validation (en utilisant des mesures physiques dérivées du banc d'essai de laboratoire). Enfin, nous terminons le chapitre en comparant la consommation électrique de l'implémentation DKF et de l'implémentation CKF sur la base des résultats des simulations. Il convient de noter que les résultats obtenus à partir des expériences présentées dans ce chapitre ont déjà été publiés dans [186], [187].

Ce chapitre est organisé comme suit :

La section 5.1 fournit les détails de l'implémentation des différents algorithmes KF qui ont été mis en œuvre dans la première série d'expériences ;

La section 5.2 décrit les configurations de simulation et de laboratoire ainsi que les expériences menées sur la plate-forme de simulation et le banc d'essai de laboratoire ;

La section 5.3 traite de l'évaluation des performances des algorithmes DKF et LKF à partir des résultats obtenus lors des simulations et des expériences physiques ;

La section 5.4 présente le profil de puissance de notre solution DKF proposée, obtenu à partir des simulations et des mesures physiques. Elle présente également les résultats des mesures physiques qui montrent comment la consommation d'énergie de la solution peut être considérablement réduite en utilisant le coprocesseur ULP du microcontrôleur ESP32 servant d'unité de traitement du nœud de capteurs ;

La section 5.5 compare la consommation d'énergie et les besoins en communication de l'algorithme DKF mis en œuvre et de l'algorithme CKF de référence dans un réseau de capteurs composé de 10 nœuds de capteurs connectés dans une topologie linéaire et simulé dans CupCarbon.

Chapitre 6 : Évaluation des performances de détection des fuites et de la consommation d'énergie des trois algorithmes DKF sélectionnés

L'objectif de ce chapitre est de mettre en œuvre les trois algorithmes DKF sélectionnés et de comparer également les résultats de leurs performances en matière de détection de fuites et de consommation d'énergie obtenus à partir de simulations et d'expériences physiques. Ceci est important car les trois algorithmes DKF sélectionnés mettent en œuvre différentes stratégies de fusion de données distribuées, avec des exigences de communication, une vitesse de convergence vers la valeur optimale du CKF et une précision d'estimation distinctes. Ces paramètres des stratégies de fusion de données distribuées mises en œuvre par ces algorithmes DKF peuvent affecter à la fois la fiabilité et la consommation d'énergie du système WWPM entièrement distribué. Par conséquent, le défi consiste à déterminer quelle solution DKF offre une meilleure performance de détection des fuites et consomme moins d'énergie.

Ce chapitre commence par une discussion sur l'implémentation des algorithmes DKF sélectionnés. Il est suivi d'une description de la configuration de simulation utilisée pour effectuer les simulations et du banc d'essai de laboratoire utilisé pour valider les résultats obtenus à partir des simulations. Nous présentons dans la suite les résultats des expériences de caractérisation des fuites réalisées sur le banc d'essai en laboratoire, où nous avons mesuré l'accélération de la surface de la conduite à différentes distances de la position de la fuite et pour différentes tailles de fuite. Aussi, nous présentons les expériences que nous avons menées sur la plateforme de simulation et le banc d'essai en laboratoire, où les performances de détection des fuites et la consommation d'énergie des algorithmes DKF ont été évaluées et comparées. Enfin, nous terminons ce chapitre en présentant les résultats de la réduction de la consommation d'énergie obtenue lors des expériences menées sur le banc d'essai de laboratoire, où nous avons mis en œuvre au niveau des nœuds de capteurs l'algorithme DKF présentant la consommation d'énergie la plus élevée avec détection hiérarchique et cycle de travail.

Ce chapitre est organisé comme suit :

La section 6.1 présente les détails de la mise en œuvre des trois algorithmes DKF sélectionnés (EDKF, ICF et SGG-ICF), qui seront évalués par le biais de simulations effectuées sur une plate-forme de simulation et d'expériences physiques réalisées sur un banc d'essai en laboratoire ;

La section 6.2 fournit une description détaillée du dispositif expérimental et des expériences menées sur la plate-forme de simulation et le banc d'essai de laboratoire ;

La section 6.3 présente les résultats de la caractérisation des fuites qui montrent comment la distance par rapport à la position de la fuite et la taille de la fuite influencent la détection des fuites en affectant les données de vibration collectées à partir de la surface de la conduite à l'aide de l'accéléromètre LSM9DS1 ;

La section 6.4 présente les résultats de simulation et de laboratoire de l'évaluation des performances des algorithmes DKF sélectionnés ;

La section 6.5 présente une évaluation de la consommation d'énergie des DKF sélectionnés à partir des résultats obtenus par des simulations et des expériences physiques ;

La section 6.6 présente la mise en œuvre du cycle d'utilisation et de la détection hiérarchique au niveau du nœud de capteur pour réduire la consommation d'énergie du nœud de capteur tout en maintenant la détection des fuites en temps réel.

Chapitre 7 : Conclusions et Perspectives

Dans ce chapitre, nous revenons tout d'abord sur les objectifs de recherche que nous nous sommes fixés, sur les résultats obtenus et sur les conclusions tirées de la réalisation des objectifs. Nous présentons également les limites de notre travail et proposons des suggestions pour les travaux futurs.

7.1 Récapitulation des objectifs, des réalisations et des conclusions de la recherche

Dans cette thèse, nous nous sommes intéressés à la détection de fuites en temps réel, de fuites fiables et de fuites à faible puissance dans les systèmes WWPM en utilisant des accéléromètres MEMS à faible coût. Notre objectif principal était d'augmenter la précision des mesures et la durée de vie du réseau grâce à une approche distribuée. Pour atteindre ce but, nous avions cinq objectifs spécifiques à réaliser. Nous présentons ci-dessous les objectifs, les réalisations et les conclusions tirées de la réalisation des objectifs.

Objectif 1 : Passer en revue les techniques de surveillance des canalisations d'eau les plus courantes disponibles dans la littérature, en mettant l'accent sur les solutions WWPM qui utilisent des capteurs de vibrations à faible coût pour la surveillance. À partir de là, identifier les lacunes dans le domaine de la surveillance des conduites en plastique à l'aide d'accéléromètres MEMS à faible coût.

Réalisation 1 : Dans le chapitre 2 de cette thèse, nous avons fait une revue complète de la littérature sur le calcul distribué dans les WSNs, les techniques de gestion de l'énergie utilisées dans les WWPM et les solutions WWPM basées sur des capteurs non-intrusifs et nous avons identifié les lacunes dans les connaissances mentionnées dans la sous-section 2.7.2 du chapitre 2.

Conclusion 1 : Dans les systèmes WWPM à grande échelle, il y a une dégradation des performances en temps réel en raison de la latence accrue résultant d'un grand nombre de communications multi-sauts. Ainsi, pour parvenir à une détection des fuites en temps réel, il est nécessaire de mettre en œuvre un calcul distribué plutôt qu'un calcul centralisé dans le système

WWPM. Récemment, il y a eu une recherche générale pour passer d'une surveillance centralisée à une surveillance distribuée dans les applications WSNs (en particulier dans les domaines de la surveillance de l'état des machines industrielles et du suivi des cibles) et cela peut également être étendu à WWPM. En outre, l'étude des techniques populaires de surveillance non intrusive montre que les accéléromètres MEMS à faible coût peuvent être utilisés pour détecter les fuites dans les tuyaux en plastique. Les accéléromètres MEMS à trois axes, plutôt que les accéléromètres MEMS à un axe, sont des candidats appropriés pour la détection des fuites dans les canalisations en plastique, mais ils nécessitent un traitement du signal supplémentaire pour augmenter la précision de la détection des fuites des systèmes WWPM qui les utilisent. En mettant en œuvre des techniques de traitement du signal qui traitent les données de manière directe dans le domaine temporel au sein des nœuds de capteurs et en impliquant la fusion de données multi-capteurs, il est probable d'augmenter les performances de détection des fuites et de réduire la consommation d'énergie.

Objectif 2 : Etablir les spécifications d'un nœud de capteurs à faible coût et à faible puissance capable d'effectuer des traitements in situ sous contraintes énergétiques. Le nœud de capteurs doit avoir une capacité de calcul suffisante pour effectuer un traitement local et être sensible à la consommation d'énergie. Sur cette base, concevoir et mettre en œuvre un nœud de capteurs sans fil basé sur les spécifications établies.

Réalisation 2 : Dans le chapitre 3 de cette thèse, nous avons effectué un examen approfondi des nœuds de capteurs commerciaux et de recherche existants en nous concentrant sur leur puissance de calcul, leur consommation d'énergie et leur coût. Le besoin d'un nœud de capteurs à faible coût, avec une puissance de calcul élevée mais une faible consommation d'énergie, nous a poussé à concevoir un nœud personnalisé plutôt que d'utiliser des nœuds de capteurs commerciaux ou de recherche existants. Notre nœud de capteurs personnalisé se compose d'un SoC ESP32 comme unité de traitement et d'un module émetteur-récepteur nRF24L01+ comme unité de communication. Nous avons choisi l'ESP32 en raison de son faible coût, de sa puissance de calcul élevée (car il intègre un processeur Xtensa LX6 32 bits à double cœur avec une vitesse de traitement allant jusqu'à 240 MHz comme cœur principal) et de sa faible consommation d'énergie (car il possède un coprocesseur ULP dont la consommation de courant est comprise entre 10 µA

et 150 µA). La puissance de calcul élevée du cœur principal de l'ESP3 a permis à notre nœud personnalisé d'effectuer un traitement local, tandis que le coprocesseur ULP lui a permis de réaliser une surveillance en temps réel tout en maintenant une faible consommation d'énergie, répondant ainsi à nos objectifs de solution WWPM en temps réel, entièrement distribuée et à faible consommation. Le module émetteur-récepteur nRF24L01+ a été sélectionné en raison de son faible coût, de son mode rafale et de sa faible consommation d'énergie (avec des courants de pointe TX/RX inférieurs à 14 mA). Pour parvenir à une détection fiable des fuites sur les tuyaux en plastique tout en maintenant une faible consommation d'énergie, nous avons proposé la mise en œuvre d'une détection hiérarchique en sélectionnant à la fois un accéléromètre de faible précision (mais de faible puissance) et un accéléromètre de haute précision. Les accéléromètres sélectionnés sont l'ADXL344 et le LSM9DS1. Nous avons choisi l'ADXL344 en raison de sa faible consommation d'énergie, de sa large bande passante et de sa capacité de détection d'événements, tandis que nous avons choisi le LSM9DS1 en raison de sa haute sensibilité et de sa faible consommation d'énergie. Nous avons également discuté de la manière de configurer le composant du nœud de capteurs pour obtenir un calcul distribué, une surveillance en temps réel et une faible consommation d'énergie. Nous avons terminé le chapitre en présentant la conception, la mise en œuvre et la configuration d'un dispositif de mesure de la puissance personnalisé pour mesurer la consommation d'énergie des nœuds de capteurs lors des expériences en laboratoire. Ce dispositif de mesure de la consommation était composé d'un INA226, d'un microcontrôleur STM nucleo-32 F303k8, d'un émetteur-récepteur nRF24L01+, d'un écran OLED 128 × 64, d'une carte SD et de deux ports USB.

Conclusion 2 : Dans une solution WWPM entièrement distribuée, les nœuds de capteurs doivent disposer d'une capacité de calcul suffisante pour effectuer un traitement local et consommer moins d'énergie, étant donné que les nœuds sont alimentés par batterie. En utilisant l'ESP32 plus fréquemment en mode ULP qu'en mode veille du modem, le nœud de capteurs a une consommation d'énergie comparable à celle des appareils bas de gamme utilisés dans les applications WSN alimentées par batterie, mais possède des ressources de calcul suffisantes pour effectuer tous les traitements nécessaires à la détection des fuites au sein du nœud de capteurs. La puissance de calcul élevée de l'ESP32 implique également qu'il lui faudra moins de temps en mode

actif pour traiter l'algorithme de détection des fuites et repasser ensuite en mode sommeil profond. Moins le nœud est en mode actif, moins il consomme d'énergie puisqu'il passe plus de temps en mode basse consommation. Aussi, la capacité de l'ESP32 à surveiller le niveau de tension de sa batterie sans nécessiter de circuit supplémentaire est avantageuse car elle permet aux nœuds de capteurs de surveiller leur consommation d'énergie en temps réel. Cette caractéristique de l'ESP32 en fait un bon candidat pour le déploiement d'applications WSN sensibles à l'énergie et nécessitant davantage de calcul au niveau du nœud, car elle permet aux nœuds de modérer leurs opérations en fonction de l'énergie restante de la batterie, ce qui les rend sensibles à l'énergie et adaptatifs. Enfin, en utilisant l'ESP32 comme unité de traitement de nos nœuds de capteurs, nous avons montré qu'il est possible d'étendre l'utilisation de l'ESP32 dans les applications WSN, étant donné que l'ESP32 a été largement utilisé dans les applications IoT dans le passé.

Objectif 3 : Proposer une solution WWPM entièrement distribuée, en temps réel et à faible consommation d'énergie, basée sur DKF pour une détection fiable des fuites dans les tuyaux en plastique. À cet égard, nous effectuerons une recherche documentaire approfondie sur les algorithmes DKF qui peuvent être mis en œuvre dans les WSN et sélectionnerons trois algorithmes DKF dont les performances de détection de fuites et la consommation d'énergie seront évaluées dans le contexte de la WWPM à l'aide d'accéléromètres MEMS à faible coût.

Réalisation 3 : Motivés par le besoin d'un algorithme à faible complexité (en termes de puissance de calcul, de stockage et de communication) au niveau des nœuds de capteurs, nous avons choisi la technologie KF pour sa mise en œuvre dans le chapitre 4 de cette thèse. Nous avons expliqué les raisons pour lesquelles nous avons choisi KF comme algorithme de traitement du signal pour la détection de fuites, en utilisant les données obtenues à partir d'accéléromètres MEMS à faible coût de chaque nœud de capteur attaché au pipeline pour répondre aux objectifs de haute fiabilité, de surveillance en temps réel et de faible consommation d'énergie. Nous avons étudié les différentes variantes de la technologie KF mises en œuvre dans les WSN. Nous avons finalement choisi d'utiliser une KF qui met en œuvre la fusion de données distribuée (DKF) plutôt qu'une KF qui met en œuvre la fusion de données centralisée (CKF). La DKF a été choisie plutôt que la CKF en raison des inconvénients de la surveillance centralisée dans les WSN, que nous avons mentionnés dans le chapitre 2 de cette thèse. Guidés par la revue des algorithmes DKF pour les

WSN à faible coût présentée par [59], nous avons sélectionné trois algorithmes DKF pour les implémenter et les évaluer dans notre solution WWPM proposée. L'un des algorithmes DKF sélectionnés est un algorithme DKF basé sur le consensus, appelé Information Consensus Filter (ICF) [64]. Cet algorithme a une bonne précision d'estimation car il converge vers la valeur optimale du CKF, et il maintient également la cohérence locale de l'estimation entre les nœuds de capteurs voisins. Cependant, il a une charge de communication élevée car chaque nœud transmet ses informations locales et reçoit des informations de tous ses voisins à chaque mise à jour de mesure, et il nécessite plusieurs itérations de consensus entre les mises à jour de mesure pour atteindre une bonne précision d'estimation. Un autre algorithme DKF sélectionné est un KF basé sur la diffusion déclenchée par les événements (EDKF) [171]. Cet algorithme a une faible charge de communication (car chaque nœud transmet ses informations locales et reçoit des informations de tous ses voisins à chaque pas de temps et avec une seule itération de communication impliquée entre les mises à jour de mesures). Cette caractéristique le rend approprié pour la mise en œuvre dans les applications WSN où il y a un délai strict avec les nœuds contraints en énergie. Cependant, la précision de l'estimation de cet algorithme est faible. Le dernier algorithme sélectionné est un algorithme DKF basé sur les commérages, appelé sample greedy gossip information consensus filter (SGG-ICF) [177]. Cet algorithme cherche à tirer le meilleur des deux mondes des DKF basés sur le consensus et sur la diffusion en obtenant une précision d'estimation élevée tout en réduisant la charge de communication. Enfin, nous avons terminé le chapitre en présentant les mesures d'évaluation couramment utilisées dans la littérature sur le WWPM pour évaluer les performances de la solution WWPM entièrement distribuée que nous proposons. Les mesures d'évaluation sélectionnées comprennent la sensibilité, la spécificité, le taux de détection des erreurs, le taux de fausses alarmes et la précision pour évaluer les performances de détection des fuites, ainsi que les mesures de courant et de tension pour la consommation d'énergie.

Conclusion 3 : La myriade d'algorithmes DKF existant dans la littérature a nécessité une étude approfondie des algorithmes DKF afin d'isoler les algorithmes DKF ayant le potentiel d'être mis en œuvre dans les WSN. Par ailleurs, la nécessité pour les nœuds de capteurs voisins d'avoir des estimations cohérentes affecte considérablement la précision de la détection des fuites d'un système WWPM entièrement distribué utilisant des accéléromètres MEMS à faible coût. La raison

en est que le traitement pour la détection des fuites est effectué au niveau des nœuds de capteurs et non au niveau d'un centre de fusion. Toute incohérence dans les estimations des nœuds de capteurs voisins entraînera des sorties contradictoires de ces derniers, ce qui conduira à des situations où un nœud déclenchera une alarme de fuite alors que son voisin n'indiquera par ailleurs aucune fuite. Pour cela, éviter de telles controverses implique de ne mettre en œuvre que des algorithmes DKF qui atteignent une cohérence locale dans leurs estimations pour la détection de fuites dans WWPM. Enfin, pour étudier l'influence d'une stratégie spécifique de fusion de données distribuées sur les performances de détection de fuites et la consommation d'énergie, il est nécessaire de sélectionner des algorithmes DKF qui atteignent une cohérence locale mais qui utilisent les différentes stratégies de fusion de données.

Objectif 4 : Démontrer l'application de DKF pour une détection fiable des fuites en temps réel dans un système WWPM qui utilise des accéléromètres MEMS à faible coût en déployant un réseau de capteurs qui met en œuvre l'un des algorithmes DKF sélectionnés. Sur cette base, évaluer les performances de détection de fuites et la consommation d'énergie de la solution distribuée proposée en effectuant des simulations et en validant les résultats de simulation à l'aide des résultats obtenus lors d'expériences physiques sur un banc d'essai de laboratoire.

Réalisation 4 : Dans le chapitre 5 de cette thèse, nous avons démontré l'application d'une solution de calcul distribué basée sur DKF pour réaliser une détection de fuites en temps réel, fiable et à faible consommation dans WWPM. La plupart des études de l'état de l'art sur le calcul distribué dans les WSN étant théoriques et validées par des simulations, nous avons dans ce chapitre validé expérimentalement la faisabilité de l'exécution du calcul distribué dans WWPM en mettant en œuvre un algorithme DKF, proposé par Battistelli et al. [171], sur un réseau de capteurs déployé sur un RDE de laboratoire pour valider les résultats de simulation. Dans la solution que nous avons proposée, les nœuds de capteurs ont effectué tous les traitements nécessaires à la détection des fuites sans avoir besoin d'une station de base centralisée pour traiter les signaux de fuite. L'objectif était d'éliminer les communications multi-sauts, de réduire la latence, de diminuer la consommation d'énergie des nœuds de capteurs et de prolonger la durée de vie du WSN. Le DKF implémenté a traité les signaux de vibration lus à partir de l'accéléromètre LSM9DS1 fixé à la surface du tuyau pour détecter l'apparition de fuites sur le pipeline. Nous avons réalisé les

simulations dans CupCarbon 4.2 et les expériences physiques sur un banc d'essai de laboratoire du réseau WSN. Nous avons effectué des simulations et des expériences physiques sur un WSN linéaire à deux nœuds, où nous avons comparé les performances de détection de fuites du DKF implémenté avec celles d'un filtre de Kalman local (LKF). Dans la suite, nous avons effectué des simulations et des expériences en laboratoire pour évaluer la consommation d'énergie de l'algorithme DKF implémenté. Enfin, nous avons effectué des simulations sur un réseau WSN global composé de dix nœuds de capteurs connectés linéairement, où nous avons comparé la consommation d'énergie du DKF implémenté avec celle du filtre de Kalman centralisé (CKF).

Conclusion 4 : Les résultats obtenus ont montré la faisabilité de l'application d'une solution entièrement distribuée pour la détection des fuites dans les WWPM. Par ailleurs, les résultats ont établi l'importance de la fusion de données distribuées dans l'amélioration de la fiabilité et de la consommation d'énergie du système de détection de fuites. En ce qui concerne l'implémentation physique, les résultats ont montré que le DKF implémenté augmente la fiabilité de la détection des fuites par rapport au LKF, tandis que les simulations sur le réseau mondial ont montré que le DKF implémenté a considérablement préservé la durée de vie du WWPM par rapport au CKF. Ces résultats indiquent que l'approche distribuée augmente les performances de détection des fuites et préserve la durée de vie du système WWPM. L'impact de cette approche n'est pas seulement limité à WWPM, mais aussi à son extension aux applications IoT critiques en termes de temps, où, au lieu d'envoyer des données brutes au cloud pour traitement et renvoi des décisions, le traitement nécessaire pour traiter les données brutes d'une application de surveillance d'événements critiques en termes de temps est effectué en utilisant les ressources informatiques embarquées des nœuds de capteurs. Cela permet de réduire la latence subie par ces applications critiques en termes de temps lors de la mise en œuvre de l'informatique centralisée. De plus, la capacité de la stratégie de fusion de données distribuée à maintenir la cohérence locale la rend très adaptée à la mise en œuvre dans des applications WSN de détection d'événements entièrement distribuées, où les nœuds de capteurs sont responsables de tout le traitement requis pour la détection d'événements. Cette caractéristique est nécessaire pour éviter les décisions contradictoires des nœuds de capteurs voisins, notamment dans les applications WSN linéaires telles que la surveillance des oléoducs et gazoducs, des ponts et tunnels ferroviaires, des frontières, des lignes électriques, etc. En

conclusion, la mise en œuvre du calcul distribué dans les applications WSN en temps réel utilisant des capteurs de faible précision qui impliquent la fusion de données distribuées des sorties locales des nœuds de capteurs voisins plutôt que la mise en œuvre du calcul distribué sans fusion de données distribuées peut améliorer les performances de surveillance.

Objectif 5 : Evaluer les performances de détection des fuites et la consommation d'énergie des algorithmes DKF sélectionnés en utilisant à la fois des simulations et des expériences physiques. Sur cette base, comparer les performances de détection de fuites et la consommation d'énergie des algorithmes DKF sélectionnés et proposer lequel des DKF est optimal pour la mise en œuvre dans les systèmes WWPM utilisant des accéléromètres MEMS à faible coût.

Réalisation 5 : Dans le chapitre 6, nous avons évalué les performances de détection des fuites et la consommation d'énergie des trois algorithmes DKF sélectionnés en effectuant des simulations et des expériences physiques sur un banc d'essai en laboratoire. Tout d'abord, nous avons effectué des expériences de caractérisation des fuites sur le banc d'essai de laboratoire, où nous avons mesuré l'accélération de la surface de la conduite pour différentes distances de la position de la fuite et différentes tailles de fuite, et nous avons dérivé l'espacement maximal entre l'accéléromètre et la position de la fuite pour une détection efficace de la fuite et la valeur de référence pour l'absence de fuite sur le pipeline. Ensuite, nous avons mis en œuvre les algorithmes DKF sélectionnés sur un WSN linéaire à deux nœuds, à la fois sur la plate-forme de simulation et sur le banc d'essai du laboratoire. Nous avons ensuite mesuré les performances de détection des fuites et la consommation d'énergie des algorithmes DKF et comparé les résultats obtenus à partir des simulations avec ceux des expériences physiques. Enfin, nous avons démontré par des expériences en laboratoire comment nous avons réduit de manière significative la consommation d'énergie de notre solution WWPM proposée en mettant en œuvre le cycle de service et la détection hiérarchique au niveau des nœuds de capteurs.

Conclusion 5 : Les résultats obtenus à partir de la caractérisation des fuites ont montré une détection efficace des fuites d'une taille supérieure à 15 L/min à une distance maximale de 2 m de la position de la fuite lorsque l'accéléromètre a été configuré pour fonctionner à la plage de détection de ±2 g. Ainsi, pour les accéléromètres MEMS avec plusieurs plages de détection, en les

configurant pour qu'ils fonctionnent à des plages de détection inférieures augmente la distance de détection des fuites. Les résultats des simulations et des expériences en laboratoire ont révélé que l'ICF présentait la meilleure performance de détection des fuites, tandis que l'EDKF présentait la plus faible performance de détection des fuites. Les résultats de la performance de détection de fuites pour EDKF dérivés des simulations étaient proches de ceux obtenus à partir des expériences en laboratoire, tandis qu'il y avait une différence significative entre la performance de détection de fuites de ICF obtenue à partir des simulations et celle dérivée des expériences en laboratoire. Cette différence s'explique par la perte de paquets dans les expériences physiques lors des communications entre les nœuds de capteurs, qui n'ont pas été prises en compte lors des simulations. Aussi, les résultats de la consommation d'énergie obtenus à partir des simulations ont révélé que la consommation d'énergie de l'EDKF était huit fois inférieure à celle de l'ICF et du SGG-ICF. Cependant, il n'y avait pas de différence significative dans les résultats de consommation d'énergie obtenus à partir des mesures physiques. C’est pourquoi, pour évaluer pleinement la consommation d'énergie des DKF, des recherches supplémentaires seront nécessaires. Néanmoins, un compromis entre les performances de détection des fuites et l'efficacité énergétique du point de vue de la simulation a révélé que l'EDKF est plus optimal dans une mise en œuvre pratique par rapport à l'ICF et au SGG-ICF car ses performances d'estimation sont moins affectées par la perte de paquets. Par ailleurs, sa faible charge de communication réduit la consommation d'énergie des nœuds de capteurs. Cet attribut fait d'EDKF un bon candidat pour la mise en œuvre dans les nœuds de capteurs alimentés par batterie. Par conséquent, plutôt que d'implémenter l'ICF ou le SGG-ICF qui ont des sensibilités plus élevées que l'EDKF mais qui ont un besoin élevé en communication, nous proposons d'utiliser des techniques d'apprentissage machine (ML) au niveau de la décision pour améliorer la sensibilité d'une solution WWPM implémentant l'EDKF. En d'autres termes, EDKF pour le filtrage au niveau de l'extraction des caractéristiques pour fournir des caractéristiques plus précises, qui sont ensuite classées par un algorithme ML au niveau de la décision pour fournir une détection fiable des fuites. Puisque cette approche implique plus de calcul que de communication, elle réduira probablement la consommation d'énergie des nœuds de capteurs, ce qui en fait un bon candidat pour la mise en œuvre dans les nœuds de capteurs alimentés par batterie utilisés dans le WWPM. C'est pourquoi, nous pouvons partiellement conclure qu'à partir d'un compromis entre la consommation d'énergie

et les performances de détection de fuites révélées par les résultats de simulation, EDKF est une meilleure solution pour la détection de fuites en temps réel dans les systèmes WWPM alimentés par batterie. Enfin, les résultats obtenus à partir de la mise en œuvre de la détection hiérarchique et du cycle de travail au niveau du nœud de capteur ont révélé une diminution significative de la consommation d'énergie, car une grande partie de l'énergie gaspillée en écoute inactive par l'émetteur-récepteur est réduite. Les résultats ont montré une réduction de la consommation d'énergie d'un facteur 8, ce qui implique que la solution proposée peut permettre une détection fiable des fuites en temps réel et, en même temps, préserver la durée de vie du système WWPM.

En résumé, notre contribution originale à la connaissance est la recherche et l'évaluation d'une solution peu coûteuse, de faible puissance et entièrement distribuée basée sur DKF pour la détection fiable en temps réel des fuites dans les systèmes WWPM en utilisant des accéléromètres MEMS peu coûteux.

7.2 Limitations et recommandations

Cette thèse présente les limites suivantes en fonction de certaines hypothèses formulées. Dans cette section, nous présentons les limites de notre étude et suggérons également des directions de recherche futures qui peuvent être explorées.

Limitation 1 : Nous avons supposé qu'il n'y avait pas de vibrations sur la canalisation dues aux demandes légitimes des clients, c'est-à-dire les vibrations résultant de l'ouverture/fermeture des robinets chez les clients. Egalement, les effets des pompes ont été négligés car nous n'avons considéré que les cas où la distribution de l'eau dans la canalisation se fait par gravité. Par ailleurs, nous avons considéré des canalisations aériennes dans nos expériences en laboratoire, mais la plupart des réseaux de distribution d'eau dans les pays en développement sont des canalisations souterraines. Enfin, notre étude s'est limitée à un WSN linéaire à deux nœuds.

Recommandation 1 : Bien que les résultats obtenus sur le banc d'essai de laboratoire soient louables, il est suggéré d'étendre nos expériences à une étude sur le terrain impliquant un véritable RFD dans des conditions réelles avec un WSN linéaire déployé à grande échelle. Ceci est important car la nature simpliste du RFD de laboratoire ne permet pas de saisir toutes les complications d'un RFD réel. Un autre point intéressant est que la plupart des études WWPM se limitent à des

simulations et des expériences sur des bancs d'essai en laboratoire. L'extension des expériences aux RDE réels contribuera largement à la littérature sur les WWPM.

Limitation 2 : Une autre limitation de cette étude est l'utilisation de la méthode de seuillage avec une valeur de base fixe pour déterminer la présence ou l'absence d'une fuite sur le pipeline. Une telle approche simpliste est plus adaptée aux systèmes de pipelines dont les caractéristiques opérationnelles sont prévisibles. Cependant, étant donné que les conditions opérationnelles des RDE réels ne sont pas constantes et peuvent changer en raison de conditions externes qui ne sont pas liées à une fuite, il est nécessaire que la valeur de base puisse varier.
Recommandation 2 : Nous suggérons la mise en œuvre de techniques d'apprentissage automatique à l'étape de décision de l'algorithme de détection des fuites. Cela implique qu'une fois que le DKF a été utilisé lors de la phase d'extraction des caractéristiques pour estimer la vibration de la surface de la conduite, la valeur peut ensuite être transmise à un classificateur entraîné lors de la phase de décision pour déterminer avec précision l'existence d'une fuite ou l'absence de fuite sur la conduite.

Limitation 3 : Cette étude se limite uniquement à la détection des fuites. Cependant, pour avoir un système WWPM complet, il est nécessaire d'intégrer la localisation des fuites.
Recommandation 3 : Explorer la mise en œuvre de techniques de localisation des fuites telles que l'analyse de la corrélation acoustique dans les expériences futures.

Limitation 4 : Le simulateur CupCarbon ne modélise que la consommation d'énergie de l'unité de communication sans tenir compte de la consommation d'énergie des unités de détection et de traitement.
Recommandation 4 : Étant donné que le simulateur CupCarbon est open source, nous suggérons pour les travaux futurs, la modification de son code source pour inclure la consommation d'énergie des unités de traitement et de détection. Cela conduira à des profils de puissance simulés des nœuds de détection proches de ceux obtenus à partir de mesures physiques. Ces profils de puissance simulés pourront alors être utilisés comme une première évaluation précise de la consommation d'énergie des algorithmes distribués avant leur implémentation physique.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks: a survey,” *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002, doi: http://dx.doi.org/10.1016/S1389-1286(01)00302-4.

[2] S. R. J. Ramson and D. J. Moni, “Applications of Wireless Sensor Networks – A Survey,” in *Proceedings of the International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*, 2017, pp. 325–329. doi: http://dx.doi.org/10.1109/ICIEEIMT.2017.8116858.

[3] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, “Applications of Wireless Sensor Networks: An Up-to-Date Survey,” *Appl. Syst. Innov.*, vol. 3, no. 1, pp. 1–24, 2020, doi: https://doi.org/10.3390/asi3010014.

[4] F. Karray, M. W. Jamal, A. Garcia-Ortiz, and A. M. Obeid, “A comprehensive survey on wireless sensor node hardware platforms,” *Computer Networks*, vol. 144, pp. 89–110, 2018, doi: https://doi.org/10.1016/j.comnet.2018.05.010.

[5] J. Jiang and C. Claudel, “A high performance, low power computational platform for complex sensing operations in smart cities,” *HardwareX*, vol. 1, pp. 22–37, 2017, doi: http://dx.doi.org/10.1016/j.ohx.2017.01.001.

[6] J. Ivković and J. L. Ivković, “Analysis of the Performance of the New Generation of 32-bit Microcontrollers for IoT and Big Data Application,” in Proceedings of the 7th International Conference on Information Society and Technology ICIST, 2017.

[7] H. Kim *et al.*, “System Architecture Directions for Post-SoC/32-bit Networked Sensors,” China, 2018. doi: https://doi.org/10.1145/ 3274783.3274839.

[8] T. Devanaboyina, B. Pillalamarri, and R. M. Garimella, “Distributed Computation in Wireless Sensor Networks: Efficient Network Architectures and Applications in WSNs,” *Int. J. Wirel. Netw. Broadband Technol.*, vol. 3, no. 4, pp. 14–32, 2015, doi: https://doi.org/10.4018/IJWNBT.2015070102.

[9] H. Huang, “Distributed Computing in Wireless Sensor Networks,” *Encyclopedia of Mobile Computing and Commerce*. pp. 202–206, 2007.

[10] M. Z. Abbas, K. A. Baker, M. Ayaz, and H. Mohamed, “Key Factors Involved in Pipeline Monitoring Techniques Using Robots and WSNs: Comprehensive Survey,” *J. Pipeline Syst. Eng. Pract.*, vol. 9, no. 2, pp. 1–15, 2018, doi: https://doi.org/10.1061/(ASCE)PS.1949-1204.0000305.

[11] F. Mieyeville, D. Navarro, O. Bareille, and M. Zielinski, “Autonomous Wireless Sensor Network for distributed active control,” in *Proceedings of the IEEE Vehicle Power and Propulsion Conference (VPPC)*, France, 2017, pp. 1–6. doi: http://dx.doi.org/10.1109/VPPC.2017.8330909.

[12] W. Zhang, B. Chen, and L. Yu, *Distributed Fusion Estimation for Sensor Networks with Communication Constraints*, 1st ed. Verlag, Singapore: Springer, 2016.

[13] S. Kartakis, W. Yu, R. Akhavan, and J. A. McCann, “Adaptive Edge Analytics for Distributed Networked Control of Water Systems,” Germany, 2016. doi: http://dx.doi.org/10.1109/IoTDI.2015.34.

[14] D. Dalta, X. Chen, T. Tsou, and S. Raghunandan, "Wireless Distributed Computing: A Survey of Research Challenges," *IEEE Communications Magazine*, vol. 50, no. 1, pp. 144–152, 2012, doi: http://dx.doi.org/10.1109/MCOM.2012.6122545.

[15] M. A. Quintana-Suárez, D. Sánchez-Rodríguez, and J. B. Alonso-Hernández, "A Low Cost Wireless Acoustic Sensor for Ambient Assisted Living Systems," *Appl. Sci.*, vol. 7, no. 9, pp. 1–15, 2017, doi: https://doi.org/10.3390/app7090877.

[16] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 7, pp. 537–568, 2009, doi: https://doi.org/10.1016/j.adhoc.2008.06.003.

[17] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, 2000, doi: http://dx.doi.org/10.1145/332833.332838.

[18] E. D. Pascale, I. Macaluso, A. Nag, M. Kelly, and L. Doyle, "The Network as a Computer: a Framework for Distributed Computing over IoT Mesh Networks," *IEEE IoT Journal*, vol. 3, no. 3, pp. 2107–2119, 2018, doi: http://dx.doi.org/10.1109/JIOT.2018.2823978.

[19] A. Petitti, D. Paola, A. Milella, M. P. Spagnolo, G. Cicirelli, and G. Attolico, "A Distributed Cooperative Architecture for Robotic Networks with Application to Ambient Intelligence," in *P. Mazzeo, P. Spagnolo, T. Moeslund (eds) Activity Monitoring by Multiple Distributed Sensing.*, 2014, pp. 1–12. doi: https://doi.org/10.1007/978-3-319-13323-2_1.

[20] G. Serpen and L. Liu, "Parallel and distributed neurocomputing with wireless sensor networks," *Neurocomputing*, vol. 172, pp. 1169–1182, 2016, doi: https://doi.org/10.1016/j.neucom.2015.08.074.

[21] R. Kacimi, R. Dhaou, and A. Beylot, "Load balancing techniques for lifetime maximizing in wireless sensor networks," *Ad Hoc Networks*, vol. 11, no. 8, pp. 2172–2186, 2013, doi: https://doi.org/10.1016/j.adhoc.2013.04.009.

[22] J. Zhang, Z. Lin, P. Tsai, and L. Xu, "Entropy-driven data aggregation method for energy-efficient wireless sensor networks," *Information Fusion*, vol. 56, pp. 103–113, 2020, doi: https://doi.org/10.1016/j.inffus.2019.10.008.

[23] B. Adelodun *et al.*, "Assessment of socioeconomic inequality based on virus-contaminated water usage in developing countries: A review," *Environmental Research*, vol. 192, pp. 1–14, 2021, doi: https://doi.org/10.1016/j.envres.2020.110309.

[24] L. Ribeiro, J. Sousa, A. S. Marques, and N. E. Simões, "Locating Leaks with TrustRank Algorithm Support," *Water*, vol. 7, no. 4, pp. 1378–1401, 2015, doi: http://dx.doi.org/10.3390/w7041378.

[25] A. R. Iyeswariya, R. M. Shamila, M. JayaLakshm, K. Maharajan, and V. Sivakumar, "A study on Water Leakage Detection in buried plastic pipes using Wireless Sensor Networks," *IJS ER*, vol. 3, no. 10, 2012.

[26] O. I. Okeya, "Detection and Localisation of Pipe Bursts in a District Metered Area Using an Online Hydraulic Model," Thesis on Doctor of Engineering, Water Engineering University of Exeter, 2018.

[27] K. B. Adedeji, Y. Haman, B. T. Abe, and M. Abu-Mahfouz, "Towards Achieving a Reliable Leakage Detection and Localization Algorithm for Application in Water Piping Networks: An Overview," *IEEE Access,* vol. 5, pp. 20272–20285, 2017, doi: http://dx.doi.org/10.1109/ACCESS.2017.2752802.

[28] UNICEF and WHO, “Progress on Sanitation and Drinking Water: 2015 Update and MDG Assessment.” UNICEF, 2015. [Online]. Available: https://www.unicef.org/publications/index_82419.html

[29] C. Bell, “The World Bank and the International Water Association to Establish a Partnership to Reduce Water Losses,” *World Bank*, 2016. http://www.worldbank.org/en/news/press-release/2016/09/01/the-world-bank-and-the-international-water-association-to-establish-a-partnership-to-reduce-water-losses. (accessed Feb. 15, 2020).

[30] African Development Bank Group, “Africa Infrastructure Knowledge Program,” 2016. http://www.afdb.org/en/ (accessed Feb. 15, 2020).

[31] Y. H. Blaise, “Suffering for water, suffering from water: access to drinking-water and associated health risks in Cameroon,” *J. Health Popul Nutr.*, vol. 28, no. 5, pp. 424–435, 2010, doi: http://dx.doi.org/10.3329/jhpn.v28i5.6150.

[32] R. Srinivasan, S. Feisso, and M. Mekonen, “NB-IoT-Based Smart Water Network,” in *Principles and Applications of Narrowband Internet of Things (NBIoT)*, Ethiopia: IGI Global, 2021, pp. 245–272. [Online]. Available: https://www.igi-global.com/chapter/nb-iot-based-smart-water-network/268953

[33] M. K. Islam and Z. Karim, “World’s Demand for Food and Water: The Consequences of Climate Change,” in *Desalination - Challenges and Opportunities*, IntechOpen, 2019.

[34] A. Ayadi, O. Ghorbel, M. S. BenSalah, and M. Abid, “A framework of monitoring water pipeline techniques based on sensors technologies,” *Journal of King Saud University – Computer and Information Sciences*, 2020, doi: https://doi.org/10.1016/j.jksuci.2019.12.003.

[35] S. El-Zahab, E. S. Abdelkader, and T. Zayed, “An accelerometer-based leak detection system,” *Mech. Syst. Signal Process.*, vol. 108, pp. 276–291, 2018, doi: https://doi.org/10.1016/j.ymssp.2018.02.030.

[36] S. El-Zahab and T. Zayed, “Leak detection in water distribution networks: an introductory overview,” *Smart Water*, vol. 4, no. 5, pp. 1–23, 2019, doi: https://doi.org/10.1186/s40713-019-0017-x.

[37] M. I. Ismail, R. A. Dziyauddin, N. A. Salleh, F. Muhammad-Sukki, N. A. Bani, and L. A. Latiff, “A Review of Vibration Detection Methods Using Accelerometer Sensors for Water Pipeline Leakage,” *IEEE Access*, vol. 7, pp. 51965–51981, 2019, doi: http://dx.doi.org/10.1109/ACCESS.2019.2896302.

[38] F. Karray, A. Garcia-Ortiz, M. W. Jamal, A. M. Obeid, and M. Abid, “EARNPIPE: A Testbed for Smart Water Pipeline Monitoring using Wireless Sensor Network,” *Procedia Computer Science*, vol. 96, pp. 285–294, 2016, doi: https://doi.org/10.1016/j.procs.2016.08.141.

[39] A. Martini, M. Troncossi, and A. Rivola, “Automatic Leak Detection in Buried Plastic Pipes of Water Supply Networks by Means of Vibration Measurements,” *Shock and Vibration*, vol. 2015, 2015, doi: http://dx.doi.org/10.1155/2015/165304.

[40] S. Rashid, H. Qaisar, H. Saeed, and E. Felemban, “A Method for Distributed Pipeline Burst and Leakage Detection in Wireless Sensor Networks Using Transform Analysis,” *Int. J. Distr. Sens. Netw.*, vol. 10, pp. 1–14, 2014, doi: https://doi.org/10.1155/2014/939657.

[41] A. M. Sadeghioon, N. Metje, D. Chapman, and C. Anthony, “SmartPipes: Smart Wireless Sensor Networks for Leak Detection in Water Pipelines,” *J. Sens. Actuator Netw.*, vol. 3, no. 1, pp. 64–78, 2014, doi: https://doi.org/10.3390/jsan3010064.

[42] I. Stoianov, L. Nachman, T. Tokmouline, and M. Csai, “PIPENET: A Wireless Sensor Network for Pipeline Monitoring,” in *Proceedings of 6th International Symposium on Information Processing in Sensor Networks*, Cambridge, MA, USA, 2007, pp. 264–273. doi: https://doi.org/10.1145/1236360.1236396.

[43] S. Tariq, Z. Hu, and T. Zayed, “Micro-electromechanical systems-based technologies for leak detection and localization in water supply networks: A bibliometric and systematic review,” *Journal of Cleaner Production*, vol. 289, pp. 1–25, 2021, doi: https://doi.org/10.1016/j.jclepro.2020.125751.

[44] L. Torres, J. Jiménez-Cabas, O. González, L. Molina, and F. López-Estrada, “Kalman Filters for Leak Diagnosis in Pipelines: Brief History and Future Research,” *J. Mar. Sci. Eng.*, vol. 8, no. 173, pp. 1–21, 2020, doi: http://dx.doi.org/10.3390/jmse8030173.

[45] M. A. Virk, M. F. Mysorewala, L. Cheded, and I. M. Ali, “Leak Detection Using Flow-Induced Vibrations in Pressurized Wall-Mounted Water Pipelines,” vol. 8, pp. 188673–188687, 2020, doi: http://dx.doi.org/10.1109/ACCESS.2020.3032319.

[46] S. Yazdekhasti, K. R. Piratla, J. Sorber, S. Atamturktur, A. Khan, and H. Shukla, “Sustainability Analysis of a Leakage-Monitoring Technique for Water Pipeline Networks,” vol. 11, no. 1, 2020, doi: https://doi.org/10.1061/(ASCE)PS.1949-1204.0000425.

[47] D. Zaman, M. K. Tiwari, A. K. Gupta, and D. Sen, “A review of leakage detection strategies for pressurised pipeline in steady-state,” *Engineering Failure Analysis*, vol. 109, pp. 1–18, 2020, doi: 10.1016/j.engfailanal.2019.104264.

[48] M. Henri, P. Carpenter, and R. E. Nicholas, *Pipeline Leak Detection Handbook*. USA: Elsevier, 2016.

[49] M. I. Ismail, R. A. Dziyauddin, and N. A. Ahmad, “Water pipeline monitoring system using vibration sensor,” in Proceedings of the EEE Conference on Wireless Sensors (ICWiSE), Malaysia, 2014. doi: http://dx.doi.org/10.1109/ICWISE.2014.7042665.

[50] C. Zhang, M. F. Lambert, M. L. Stephens, J. Gong, and B. S. Cazzolato, “Pipe crack early warning for burst prevention by permanent acoustic noise level monitoring in smart water networks,” *Urban Water Journal*, pp. 1–12, 2020.

[51] H. Mustafa and P. H. Chou, “Embedded Damage Detection in Water Pipelines Using Wireless Sensor Networks,” in *Proceedings of the 2012 IEEE 14th International Conference on High Performance Computing and Communications*, 2012, pp. 1578–1586. doi: http://dx.doi.org/10.1109/HPCC.2012.230.

[52] R. P. Evans, J. D. Blotter, and A. G. Stephens, “Flow Rate Measurements Using Flow-Induced Pipe Vibration,” *J. Fluids Eng.*, vol. 126, no. 2, pp. 280–285, 2004, doi: https://doi.org/10.1115/1.1667882.

[53] G. Nwalozie and A. Azubogu, “Design and Implementation of Pipeline Monitoring System using Acceleration-Based Wireless Sensor Network,” *IJE*, vol. 3, no. 9, pp. 49–58, 2014.

[54] K. Marmarokopos, D. Doukakis, G. Frantziskonis, and M. Avloniti, “Leak Detection in Plastic Water Supply Pipes with a High Signal-to-Noise Ratio Accelerometer,” *Mea. & Cont.*, vol. 51, no. 1 & 2, pp. 27–37, 2018, doi: https://doi.org/10.1177/0020294018758526.

[55] S. Hamilton and B. Charalambous, *Leak Detection: Technology and Implementation*, 2nd ed. London, UK: IWA Publishing, 2020.

[56] M. J. Brennan, D. N. Chapman, P. F. Joseph, N. Metje, J. M. Muggleton, and E. Rustighi, *Achieving Zero Leakage by 2050: Leak Detection and Location Methods - Acoustic Leak*

Detection. Room EA1, 1-7 Great George Street, London SW1P 3AA , UK: UK Water Industry Research Limited, 2017.

[57] U. Baroudi, A. Al-Roubaiey, and A. Devendiran, "Pipeline Leak Detection Systems and Data Fusion: A Survey," *IEEE Access*, vol. 7, pp. 97426–97439, 2019, doi: http://dx.doi.org/10.1109/ACCESS.2019.2928487.

[58] M. I. Ismail, R. A. Dziyauddin, N. A. Salleh, R. Ahmad, M. H. Azmi, and H. M. Kaidi, "Analysis and Procedures for Water Pipeline Leakage Using Three-Axis Accelerometer Sensors: ADXL335 and MMA7361," *IEEE Access*, vol. 6, pp. 71249–71261, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2878862.

[59] S. He, H. Shin, S. Xu, and A. Tsourdos, "Distributed estimation over a low-cost sensor network: A Review of state-of-the-art," *Information Fusion*, vol. 54, pp. 21–43, 2020, doi: https://doi.org/10.1016/j.inffus.2019.06.026.

[60] M. S. Mahmoud and Y. Xia, *Networked Filtering and Fusion in Wireless Sensor Networks*. CRC Press, 2015.

[61] M. Elleuchi, M. Boujeleben, and M. Abid, "Energy-efficient routing model for water pipeline monitoring based on wireless sensor networks," *Int'l J. of Computers & Appl.*, 2019, doi: http://dx.doi.org/10.1080/1206212X.2019.1682239.

[62] Y. Liu, X. Ma, Y. Li, Y. Tie, Y. Zhang, and J. Gao, "Water Pipeline Leakage Detection Based on Machine Learning and Wireless Sensor Networks," *Sensors*, vol. 19, no. 23, 2019, doi: http://dx.doi.org/10.3390/s19235086.

[63] M. Taj and A. Cavallaro, "Distributed and Decentralized multicamera tracking," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 44–58, 2011, doi: http://dx.doi.org/10.1109/MSP.2011.940281.

[64] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus filters and their application in distributed camera networks," *IEEE Trans. on Auto Control*, vol. 58, no. 12, pp. 3112–3125, 2013, doi: http://dx.doi.org/10.1109/TAC.2013.2277621.

[65] M. Abdulkarem, K. Samsudin, F. Z. Rokhani, and M. F. Rasid, "Wireless sensor network for structural health monitoring: A contemporary review of technologies, challenges, and future direction," *Structural Health Monitoring*, vol. 19, no. 3, pp. 1–43, 2020, doi: https://doi.org/10.1177/1475921719854528.

[66] P. Costa, L. Mottola, A. L. Murphy, and G. P. Picco, "TeenyLIME: transiently shared tuple space middleware for wireless sensor networks," in *Proceedings of the international workshop on middleware for sensor networks*, 2006, pp. 43–48. doi: http://dx.doi.org/10.1145/1176866.1176874.

[67] D. Oliveira, M. Costa, S. Pinto, and T. Gomes, "The Future of Low-End Motes in the Internet of Things: A Prospective Paper," *electronics*, vol. 9, no. 1, 2020, doi: https://doi.org/10.3390/electronics9010111.

[68] C. Guyeux, M. Haddad, M. Hakem, and M. Lagacherie, "Efficient distributed average consensus in wireless sensor networks," *Computer Communications*, pp. 115–121, 2020, doi: https://doi.org/10.1016/j.comcom.2019.11.006.

[69] O. O. Olakanmi and A. Dada, "Wireless Sensor Networks (WSNs): Security and Privacy Issues and Solutions," in *Wireless Mesh Networks - Security, Architectures and Protocols*, IntechOpen, 2020. [Online]. Available: https://www.intechopen.com/books/wireless-mesh-networks-security-architectures-and-protocols/wireless-sensor-networks-wsns-security-and-privacy-issues-and-solutions

[70] H. Yetgin, K. T. Cheung, M. El-Hajjar, and L. Hanzo, “A Survey of Network Lifetime Maximization Techniques in Wireless Sensor Networks,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 828–854, 2017, doi: http://dx.doi.org/10.1109/COMST.2017.2650979.

[71] F. Engmann, F. A. Katsriku, J. Abdulai, K. S. Adu-Manu, and F. K. Banaseka, “Prolonging the Lifetime of Wireless Sensor Networks: A Review of Current Techniques,” *Wireless Comm. & Mobile Computing*, vol. 2018, pp. 1–23, 2018, doi: https://doi.org/10.1155/2018/8035065.

[72] M. Prauzek, J. Konecny, M. Borova, K. Janosova, J. Hlavica, and P. Musilek, “Energy Harvesting Sources, Storage Devices and System Topologies for Environmental Wireless Sensor Networks: A Review,” *Sensors*, vol. 18, no. 8, pp. 1–12, 2018, doi: https://doi.org/10.3390/s18082446.

[73] G. Peruzzi and A. Pozzebon, “A Review of Energy Harvesting Techniques for Low Power Wide Area Networks (LPWANs),” vol. 13, no. 13, pp. 1–24, 2020, doi: https://doi.org/10.3390/en13133433.

[74] M. Habibzadeh, M. Hassanalieragh, A. Ishikawa, T. Soyata, and G. Sharma, “Hybrid Solar-Wind Energy Harvesting for Embedded Applications: Supercapacitor-Based System Architectures and Design Tradeoffs,” *IEEE Circuits Syst. Mag.*, vol. 17, no. 4, pp. 29–63, 2017, doi: http://dx.doi.org/10.1109/MCAS.2017.2757081.

[75] P. Azad and V. Sharma, “Cluster Head Selection in Wireless Sensor Networks under Fuzzy Environment,” *International Scholarly Research Notices*, vol. 2013, 2013, doi: https://doi.org/10.1155/2013/909086.

[76] B. Nazir and H. Hasbullah, “Mobile Sink based Routing Protocol (MSRP) for Prolonging Network Lifetime in Clustered Wireless Sensor Network,” in Proceedings of the 2010 International Conference on Computer Applications and Industrial Electronics, Kuala Lumpur, Malaysia, 5-8 Dec. 2010. doi: http://dx.doi.org/10.1109/ICCAIE.2010.5735010.

[77] A. M. Obeid, N. Atitallah, K. Loukil, M. Abid, and M. Bensalah, “A Survey on Efficient Power Consumption in Adaptive Wireless Sensor Networks,” *Wireless Pers Commun*, vol. 101, pp. 101–117, 2018, doi: https://doi.org/10.1007/s11277-018-5678-5.

[78] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-Efficient Communication Protocol for Wireless Microsensor Networks,” in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Maui, HI, USA, USA, Jan. 2000, pp. 1–10. doi: http://dx.doi.org/10.1109/HICSS.2000.926982.

[79] A. Musaddiq, Y. B. Zikria, Zulqarnain, and S. W. Kim, “Routing protocol for Low-Power and Lossy Networks for heterogeneous traffic network,” *EURASIP J. Wirel. Commun. Netw.*, vol. 2020, no. 1, p. 21, 2020, doi: http://dx.doi.org/10.1186/s13638-020-1645-4.

[80] S. A. Fatemifar and R. Javidan, “A new load balancing clustering method for the RPL protocol,” *Telecommun. Syst.*, 2021, doi: http://dx.doi.org/10.1007/s11235-021-00760-7.

[81] X. Zhang, X. Lu, and X. Zhang, “Mobile wireless sensor network lifetime maximization by using evolutionary computing methods,” *Ad Hoc Netw.*, vol. 101, p. 102094, 2020, doi: 10.1016/j.adhoc.2020.102094.

[82] F. C. Delicato and P. F. Pires, “Energy awareness and efficiency in wireless sensor networks: From physical devices to the communication link,” in *Energy-Efficient Distributed Computing Systems*, John Wiley & Sons, 2012.

[83] A. Tripathi, S. Gupta, and B. Chourasiya, “Survey on data aggregation techniques for wireless sensor networks,” *IJARCCE*, vol. 3, no. 7, pp. 7366–7371, 2014.

[84] G. Feng, “Optimisation of vibration monitoring nodes in wireless sensor networks,” Doctoral thesis, University of Huddersfield, UK, 2016. [Online]. Available: http://eprints.hud.ac.uk/id/eprint/30320/

[85] H. Elazhary, “Internet of Things (IoT), mobile cloud,cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions,” *J. of Net & Comp. Appl.*, vol. 128, pp. 105–140, 2019, doi: https://doi.org/10.1016/j.jnca.2018.10.021.

[86] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, “Edge computing: A survey,” *Fut. Gen. Comp. Sys.*, vol. 97, pp. 219–235, 2019, doi: https://doi.org/10.1016/j.future.2019.02.050.

[87] C. Bormann, M. Ersue, and A. Keranen, “Terminology for constrained node networks,” *RFC 7228 (Informational), Internet Engineering Task Force*, May 2014. https://tools.ietf.org/html/rfc7228 (accessed Jul. 29, 2020).

[88] O. Hahm, E. Baccelli, H. Petersen, and N. Tsiftes, “Operating Systems for Low-End Devices in the Internet of Things: A Survey,” *IEEE Internet Things J.*, vol. 3, no. 5, p. 720 734, 2016, doi: http://dx.doi.org/10.1109/JIOT.2015.2505901.

[89] M. O. Ojo, S. Giordano, G. Procissi, and I. N. Seitanidis, “A Review of Low-End, Middle-End, and High-End Iot Devices,” *IEEE Access*, vol. 6, pp. 70528–70554, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2879615.

[90] M. Capra, R. Peloso, G. Masera, M. R. Roch, and M. Martina, “Edge Computing: A Survey on the Hardware Requirements in the Internet of Things World,” *Future Internet*, vol. 11, no. 4, p. p.100, 2019, doi: https://doi.org/10.3390/fi11040100.

[91] C. Jiang, T. Fan, H. Gao, W. Shi, and L. Liu, “Energy aware edge computing: A survey,” *Computer Communications*, vol. 151, p. Computer Communications, 2020, doi: http://dx.doi.org/10.1016/j.comcom.2020.01.004.

[92] M. Štula, D. Stipaničev, and L. Šerić, “Multi-Agent Systems in Distributed Computation,” in *proceedings of 6th KES International Conference*, Croatia, 2012, pp. 629–636. doi: https://doi.org/10.1007/978-3-642-30947-2_68.

[93] T. Ramji, B. Ramkumar, and M. S. Manikandan, “Resource and Subcarriers Allocation for OFDMA based Wireless Distributed Computing System,” in *Proceedings of the IEEE International Advance Computing Conference (IACC)*, India, 2014, pp. 338–342. doi: http://dx.doi.org/10.1109/IAdCC.2014.6779345.

[94] C. F. Chiasserini, “On the Concept of Distributed Digital Signal Processing in Wireless Sensor Networks,” in *proceedings of IEEE Military Communication Conference (MILCOM)*, CA, USA, 2002, vol. 1, pp. 260–264. doi: http://dx.doi.org/10.1109/MILCOM.2002.1180450.

[95] G. Feng, J. Gu, D. Zhen, M. Aliwan, F. Gu, and A. D. Ball, “Implementation of Envelope Analysis on a Wireless Condition Monitoring System for Bearing Fault Diagnosis,” *I. J. of Auto. & Comp.*, vol. 12, no. 1, pp. 14–24, 2015, doi: http://dx.doi.org/10.1007/s11633-014-0862-x.

[96] N. Dziengel, M. Seiffert, M. Ziegert, S. Adler, S. Pfeiffer, and J. Schiller, “Deployment and evaluation of a fully applicable distributed event detection system in Wireless Sensor

Networks," *Ad Hoc Networks*, vol. 37, pp. 160–182, 2016, doi: http://dx.doi.org/10.1016/j.adhoc.2015.08.017.

[97] P. Alriksson and A. Rantzer, "Experimental Evaluation of a Distributed Kalman Filter Algorithm," in Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, Dec. 2007. doi: http://dx.doi.org/10.1109/CDC.2007.4434590.

[98] C. Titouna, F. Naït-Abdesselam, and A. Khokhar, "DODS: A Distributed Outlier Detection Schemefor Wireless Sensor Networks," *Computer Networks*, vol. 161, pp. 93–101, 2019, doi: https://doi.org/10.1016/j.comnet.2019.06.014.

[99] M. A. Adegboye, W. Fung, and A. Karnik, "Recent Advances in Pipeline Monitoring and Oil Leakage Detection Technologies: Principles and Approaches," *Sensors*, vol. 19, no. 2548, pp. 1–36, 2019, doi: http://dx.doi.org/10.3390/s19112548.

[100] T. K. Chan, C. S. Chin, and X. Zhong, "Review of Current Technologies and Proposed Intelligent Methodologies for Water Distributed Network Leakage Detection," *IEEE Access*, vol. 6, pp. 78846–78867, 2018, doi: http://dx.doi.org/10.1109/ACCESS.2018.2885444.

[101] Y. He, S. Li, and Y. Zheng, "Distributed State Estimation for Leak Detection in Water Supply Networks," *IEEE/CAA Journal of Automatica Sinica,* pp. 1–9, 2017, doi: http://dx.doi.org/10.1109/JAS.2017.7510367.

[102] A. Navarro, O. Begovich, J. D. S. Torres, and G. Besancon, "Leak Detection and Isolation Using an Observer Based on Robust Sliding Mode Differentiators," in *Proceedings of the World Automation Congress (WAC)*, 2012, pp. 1–6.

[103] L. Torres, G. Besançon, and C. Verde, "Leak detection using parameter identification," in *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, 2012, vol. 45, pp. 910–915. doi: https://doi.org/10.3182/20120829-3-MX-2028.00070.

[104] A. Lay-Ekuakille, P. Vergallo, and A. Trotta, "Impedance Method for Leak Detection in Zigzag Pipelines," *Measurement Science Review*, vol. 10, no. 6, pp. 208–212, 2010, doi: http://dx.doi.org/10.2478/v10048-010-0036-0.

[105] J. C. Martins and P. Seleghim, "Assessment of the performance of acoustic and mass balance methods for leak detection in pipelines for transporting liquids," *J. Fluids Eng.*, vol. 32, no. 1, 2010, doi: http://dx.doi.org/10.1115/1.4000736.

[106] J. Smith, J. Chae, S. Learn, and R. Hugo, "Pipeline rupture detection using real-time transient modelling and convolutional neural networks," Calgary, Alberta, Canada, 2018. doi: https://doi.org/10.1115/IPC2018-78426.

[107] M. Mysorewala, M. Sabih, L. Cheded, and M. T. Nasir, "A Novel Energy-Aware Approach for Locating Leaks in Water Pipeline Using a Wireless Sensor Network and Noisy Pressure Sensor Data," *Int'l J. of Distr. Sensor Nets*, pp. 1–10, 2015, doi: https://doi.org/10.1155/2015/675454.

[108] S. Kim, "Inverse Transient Analysis for a Branched Pipeline System with Leakage and Blockage Using Impedance Method," *Procedia Engineering*, vol. 89, pp. 1350–1357, 2014, doi: https://doi.org/10.1016/j.proeng.2014.11.456.

[109] P. J. Lee, M. F. Lambert, A. R. Simpson, J. P. Vítkovsky, and D. Misiunas, "Leak location in single pipelines using transient reflections.," *Australian Journal of Water Resources*, vol. 11, no. 1, pp. 53–65, 2007, doi: https://doi.org/10.1080/13241583.2007.11465311.

[110] X. Wang, M. F. Lambert, and A. R. Simpson, “Detection and location of a partial blockage in a pipeline using damping of fluid transients.,” *Journal of Water Resources Planning and Management*, vol. 13, no. 3, pp. 244–249, 2005, doi: https://doi.org/10.1061/(ASCE)0733-9496(2005)131:3(244).

[111] A. P. Wijayanto and I. M. Maisa, “Effects of Variations in Pipe Leakage Diameter and Pipe Leakage Distance to the FFT Diagrams,” *Journal of Physics: Conference Series*, pp. 1–6, 2019, doi: http://dx.doi.org/10.1088/1742-6596/1351/1/012055.

[112] S. E. Christodoulou, E. Kourti, and A. Agathokleous, “Waterloss Detection in Water Distribution Networks using Wavelet Change-Point Detection,” *Water Resour Manage*, vol. 31, pp. 979–994, 2017, doi: https://doi.org/10.1007/s11269-016-1558-5.

[113] M. Nicola, C. Nicola, A. Vintilă, I. Hurezeanu, and M. Duță, “Pipeline Leakage Detection by Means of Acoustic Emission Technique Using Cross-Correlation Function,” *J. of Mech. Eng. & Auto.*, vol. 8, no. 2, pp. 59–67, 2018, doi: http://dx.doi.org/10.5923/j.jmea.20180802.03.

[114] S. Rashid, U. Akram, and S. A. Khan, “WML: Wireless Sensor Network based Machine Learning for Leakage Detection and Size Estimation,” in *Proceedings of the 6th International Conferenceon Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2015)*, 2015, vol. 63, pp. 171–176.

[115] T. B. Quy, S. Muhammad, and J. Kim, “A Reliable Acoustic EMISSION Based Technique for the Detection of a Small Leak in a Pipeline System,” *energies*, vol. 12, no. 8, pp. 1–18, 2019, doi: https://doi.org/10.3390/en12081472.

[116] J. Bohorquez, A. R. Simpson, M. F. Lambert, and B. Alexander, “Merging Fluid Transient Waves and Artificial Neural Networks for Burst Detection and Identification in Pipelines,” *J. Water Resour. Plann. Manage*, vol. 147, no. 1, pp. 1–14, 2021, doi: http://dx.doi.org/10.1061/(ASCE)WR.1943-5452.0001296. T.

[117] E. J. Pérez-Pérez, F. R. López-Estrada, G. Valencia-Palomo, L. Torres, and V. Puig, “Leak diagnosis in pipelines using a combined artificial neural network approach,” *104677*, vol. 107, pp. 1–10, 2021, doi: https://doi.org/10.1016/j.conengprac.2020.104677.

[118] X. Xu and B. Karney, “An Overview of Transient Fault Detection Techniques,” in *Modeling and Monitoring of Pipelines and Networks*, Springer, 2017.

[119] T. R. Sheltami, A. Bala, and E. M. Shakshuki, “Wireless sensor networks for leak detection in pipelines: a survey,” *J Ambient Intell Human Comput*, vol. 7, pp. 347–356, 2016, doi: http://dx.doi.org/10.1007/s12652-016-0362-7.

[120] M. Murad, A. A. Sheikh, M. A. Manzoor, E. Felemban, and S. Qaisar, “A survey on current underwater acoustic sensor network applications,” *Int. J. Comput. Theory Eng.*, vol. 7, no. 1, pp. 51–55, 2015, doi: http://dx.doi.org/10.7763/IJCTE.2015.V7.929.

[121] L. Wong *et al.*, “Leak detection and quantification of leak size along water pipe using optical fibre sensors package,” *Electron. J. Struct. Eng*, vol. 18, pp. 47–53, 2018.

[122] A. Cataldo, E. De Benedetto, G. Cannazza, G. Leucci, L. De Giorgi, and C. Demitri, “Enhancement of leak detection in pipelines through time-domain reflectometry/ground penetrating radar measurements.,” *IET Sci. Meas. Tech.*, vol. 1, no. 6, pp. 696–702, 2017, doi: http://dx.doi.org/10.1049/iet-smt.2016.0310.

[123] Y. Shi, C. Zhang, R. Li, M. Cai, and G. Jia, “Theory and Application of Magnetic Flux Leakage Pipeline Detection,” *Sensors*, vol. 15, no. 12, pp. 31036–31055, 2015, doi: http://dx.doi.org/10.3390/s151229845.

[124] M. J. Brennan, Y. Gao, P. C. Ayala, F. C. Almeida, and P. F. Joseph, “Amplitude distortion of measured leak noise signals caused by instrumentation: Effects on leak detection in water pipes using the cross-correlation method,” *Journal of Sound and Vibration*, vol. 461, pp. 1–16, 2019, doi: https://doi.org/10.1016/j.jsv.2019.11490.
[125] F. Almeida, M. Brennan, P. Joseph, S. Whitfield, S. Dray, and A. Paschoalini, “On the Acoustic Filtering of the Pipe and Sensor in a Buried Plastic Water Pipe and its Effect on Leak Detection: An Experimental Investigation,” vol. 14, no. 3, pp. 5595–5610, 2014, doi: https://doi.org/10.3390/s140305595.
[126] O. Hunaidi, A. Wang, M. Bracken, T. Gambino, and C. Fricke, “Acoustic Methods for Locating Leaks in Municipal Water Pipe Networks,” in *Proceedings of the International Water Demand Management Conference*, Dead Sea - Jordan, 2004, pp. 1–14.
[127] F. Okosun, P. Cahill, B. Hazra, and V. Pakrashi, “Vibration-based leak detection and monitoring of water pipes using output-only piezoelectric sensors,” *Eur. Phys. J. Special Topics*, pp. 1659–1675, 2019, doi: https://doi.org/10.1140/epjst/e2019-800150-6.
[128] J. P. Amezquita-Sanchez and H. Adeli, “Signal Processing Techniques for Vibration-Based Health Monitoring of Smart Structures,” *Arch Computat Methods Eng*, vol. 23, pp. 1–15, 2016, doi: https://doi.org/10.1007/s11831-014-9135-7.
[129] Anderiansyah and I. M. Miasa, “Comparative Study of Vibration Signal Processing on Pipe Leak Case,” *J. Phys.: Conf. Ser.*, vol. 1351, 2019, doi: http://dx.doi.org/10.1088/1742-6596/1351/1/012012.
[130] M. M. Taha, A. Noureldin, J. L. Lucero, and T. J. Baca, “Wavelet Transform for Structural Health Monitoring: A Compendium of Uses and Features,” *Structural Health Monitoring*, vol. 5, no. 3, pp. 267–295, 2006, doi: https://doi.org/10.1177/1475921706067741.
[131] L. L. Ting, J. Y. Tey, A. C. Tan, Y. J. King, and F. A. Rahman, “Water leak location based on improved dual-tree complex wavelet transform with soft thresholding de-noising,” *Applied Acoustic*, 2020, doi: https://doi.org/10.1016/j.apacoust.2020.107751.
[132] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*, 5th ed. Hoboken, New Jersey, US: John Wiley & Sons, 2015.
[133] A. Entezami, H. Shariatmadar, and S. Mariani, “Early damage assessment in large-scale structures by innovative statistical pattern recognition methods based on time series modeling and novelty detection,” *Advances in Engineering Software*, vol. 150, 2020, doi: https://doi.org/10.1016/j.advengsoft.2020.102923.
[134] C. A. Junior, J. Luiz de Medeiros, and O. F. Araújo, “ARX modeling approach to leak detection and diagnosis,” *Journal of Loss Prevention in the Process Industries*, vol. 23, no. 3, pp. 462–475, 2010, doi: https://doi.org/10.1016/j.jlp.2010.03.001.
[135] G. Wang, D. Dong, and C. Fang, “Leak detection for transport pipelines based on autoregressive modeling,” *IEEE Transactions on Instrumentation and Measurement*, vol. 42, no. 1, pp. 68–71, 1993, doi: http://dx.doi.org/10.1109/19.206686.
[136] R. Wu and M. R. Jahanshahi, “Data fusion approaches for structural health monitoring and system identification: Past, present, and future,” *Structural Health Monitoring*, vol. 19, no. 2, pp. 1–35, 2018, doi: http://dx.doi.org/10.1177/1475921718798769.
[137] J. Choi, J. Shin, C. Song, S. Han, and D. I. Park, “Leak Detection and Location of Water Pipes Using Vibration Sensors and Modified ML Prefilter,” *Sensors*, vol. 17, no. 2104, pp. 1–17, 2017, doi: https://doi.org/10.3390/s17092104.

[138] R. A. Cody and S. Narasimhan, “A field implementation of linear prediction for leak-monitoring in water distribution networks,” *Advanced Engineering Informatics*, vol. 45, pp. 1–13, 2020, doi: https://doi.org/10.1016/j.aei.2020.101103.
[139] M. Mysorewala, “Time and Energy Savings in Leak Detection in WSN-Based Water Pipelines: A Novel Parametric Optimization-Based Approach,” *Water Resources Management*, vol. 33, pp. 2057–2071, 2019, doi: https://doi.org/10.1007/s11269-019-02232-9.
[140] N. Saqib, M. F. Mysorewala, and L. Cheded, “A novel multi-scale adaptive sampling-based approach for energy saving in leak detection for WSN-based water pipelines,” *Measurement Science and Technology*, vol. 28, no. 12, pp. 1–27, 2017, doi: http://dx.doi.org/10.1088/1361-6501/aa8a2a.
[141] I. Jawhar, N. Mohamed, J. Al-Jaroodi, and S. Zhang, “A Framework for Using Unmanned Aerial Vehicles for Data Collection in Linear Wireless Sensor Networks,” *J Intell Robot Sys*, vol. 74, pp. 437–453, 2014, doi: http://dx.doi.org/10.1007/s10846-013-9965-9.
[142] H. Ali and J. Choi, “A Review of Underground Pipeline Leakage and Sinkhole Monitoring Methods Based on Wireless Sensor Networking,” *Sustainability*, vol. 11, no. 15, pp. 1–24, 2019, doi: https://doi.org/10.3390/su11154007.
[143] M. A. M. Vieira, A. B. da Cunha, and D. C. da Silva, “Designing Wireless Sensor Nodes,” in *Proceedings of the Embedded Computer Systems: Architectures, Modeling, and Simulation*, Berlin, Heidelberg, 2006, pp. 99–108. doi: http://dx.doi.org/10.1007/11796435_12.
[144] H. Karl and A. Willig, *Protocols and Architectures Wireless Sensor Networks*. John Wiley & Sons, 2005.
[145] M. Obaidat and S. Misra, “Inside a wireless sensor node: structure and operations,” in *Principles of Wireless Sensor Networks*, Cambridge: Cambridge University Press, 2014, pp. 14–29.
[146] R. Bischoff, J. Meyer, and G. Feltrin, “Wireless Sensor Network Platforms,” in *Encyclopedia of Structural Health Monitoring*, John Wiley & Sons, 2009.
[147] F. Zhao and L. J. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, 1st ed. 2005.
[148] R. Fisher, L. Ledwaba, G. Hancke, and C. Kruger, “Open Hardware: A Role to Play in Wireless Sensor Networks,” *Sensors*, vol. 15, no. 3, pp. 6818–6844, 2015, doi: http://dx.doi.org/10.3390/s150306818.
[149] Brüel & Kjær company, “Accelerometers for Vibration Measurement,” *Accelerometers - TEDS Accelerometer | Brüel & Kjær*. https://www.bksv.com/en/transducers/vibration/accelerometers (accessed May 05, 2021).
[150] M. Galetto, A. Schiavi, G. Genta, A. Prato, and F. Mazzoleni, “Uncertainty evaluation in calibration of low-cost digital MEMS accelerometers for advanced manufacturing applications,” *CIRP Annals - Manufacturing Technology*, vol. 68, no. 1, pp. 535–538, 2019, doi: https://doi.org/10.1016/j.cirp.2019.04.097.
[151] R. R. Ribeiro and R. M. Lameiras, “Evaluation of low-cost MEMS accelerometers for SHM: frequency and damping identification of civil structures,” *Lat. Am. j. solids struct.*, vol. 16, no. 7, 2019, doi: https://doi.org/10.1590/1679-78255308.

[152] M. Varanis, A. Silva, A. Mereles, and R. Pederiva, “MEMS accelerometers for mechanical vibrations analysis: a comprehensive review with applications,” *J Braz. Soc. Mech. Sci. Eng.*, vol. 40, 2018, doi: https://doi.org/10.1007/s40430-018-1445-5.
[153] O. Scussel, M. J. Brennan, F. C. Almeida, J. M. Muggleton, E. Rustighi, and P. F. Joseph, “Estimating the spectrum of leak noise in buried plastic water distribution pipes using acoustic or vibration measurements remote from the leak,” *Mech. Sys & Sig. Proc.*, vol. 147, pp. 1–13, 2021, doi: https://doi.org/10.1016/j.ymssp.2020.107059.
[154] Pycom, “Pycom go invent,” *Pycom go invent*. https://pycom.io/ (accessed Jul. 29, 2020).
[155] CircuitPython, “The easiest way to program microcontrollers.” https://circuitpython.org/ (accessed Jul. 29, 2020).
[156] E. Baccelli, O. Hahm, M. Günes, M. Wählisch, and T. C. Schmidt, “RIOT OS: Towards an OS for the Internet of Things,” in *Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Turin, Italy, 2013, p. 79-80. doi: http://dx.doi.org/10.1109/INFCOMW.2013.6970748.
[157] Y. B. Zikria, S. W. Kim, O. Hahm, M. K. Afzal, and M. Y. Aalsalem, “Internet of Things (IoT) Operating Systems Management: Opportunities, Challenges, and Solution,” *Sensors*, vol. 19, no. 8, p. p.1793, 2019, doi: http://dx.doi.org/10.3390/s19081793.
[158] Expressif Systems, “ESP32 SoC.” https://www.espressif.com/en/products/hardware/esp32/overview (accessed Feb. 15, 2020).
[159] Nordic Semiconductor, “nRF24L01+ Single Chip 2.4GHz Transceiver Product Specification v1.0,” Sep. 2008. https://www.sparkfun.com/datasheets/Wireless/Nordic/nRF24L01P_Product_Specification_1.0 (accessed Feb. 15, 2020).
[160] H. Saha, S. Mandal, S. Mintra, S. Banerjee, and U. Saha, “Comparative Performance Analysis between nRF24L01+ and XBEE ZB Module Based Wireless Ad-hoc Networks,” *I. J. Computer Network and Information Security*, vol. 7, pp. 36–44, 2017, doi: http://dx.doi.org/10.5815/ijcnis.2017.07.05.
[161] STMicroelectronics, “LSM9DS1 Datasheet, DocID025715 Rev 2,” Nov. 2014. https://www.st.com/resource/en/datasheet/DM00103319, (accessed Feb. 15, 2020).
[162] R. Ge and K. W. Cameron, “Power-aware high-performance computing,” in *Energy-Efficient Distributed Computing Systems*, John Wiley & Sons, 2012, p. 830.
[163] B. Giovino, “Making sense of current sensing,” Mouser Electronics, White paper. [Online]. Available: https://www.mouser.cn/pdfdocs/CurrentSenseWhitePaper-2.PDF
[164] E. F. Nakamura, A. F. Loureiro, and A. C. Frey, “Information Fusion for Wireless Sensor Networks: Methods, Models, and Classifications,” *ACM Computing Surveys*, vol. 39, no. 3, 2007, doi: https://doi.org/10.1145/1267070.1267073.
[165] J. A. Delgado-Aguiñaga, G. Besançon, O. Begovich, and J. E. Carvajal, “Multi-leak diagnosis in pipelines based on Extended Kalman Filter,” *Control Eng. Pract.*, vol. 49, pp. 139–148, 2016, doi: http://dx.doi.org/10.1016/j.conengprac.2015.10.008.
[166] M. B. Rhudy, R. A. Salguero, and K. Holappa, “A Kalman Filtering Tutorial for Undergraduate Students,” *IJCSES*, vol. 8, no. 1, 2017, doi: http://dx.doi.org/10.5121/ijcses.2017.8101.
[167] X. Zhang and Y. Shen, “Distributed Kalman Filtering Based on the Non-Repeated Diffusion Strategy,” *Sensors*, vol. 20, no. 23, pp. 1–16, 2020, doi: https://doi.org/10.3390/s20236923.

[168] C. Chong, K. Chang, and S. Mori, “A Review of Forty Years of Distributed Estimation,” in *proceedings of the 21st International Conference on Information Fusion (FUSION)*, Cambridge, UK, Jul. 2018, pp. 70–77. doi: http://dx.doi.org/10.23919/ICIF.2018.8455318.
[169] M. S. Mahmoud and H. M. Khalid, “Distributed Kalman filtering: a bibliographic review,” *IET Control Theory Appl*, vol. 7, no. 4, pp. 483–501, 2013, doi: http://dx.doi.org/10.1049/iet-cta.2012.0732.
[170] R. Olfati-Saber, “Distributed Kalman Filter with Embedded Consensus Filters,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, Dec. 2005, pp. 8179-8184. doi: http://dx.doi.org/10.1109/CDC.2005.1583486.
[171] G. Battistelli, L. Chisci, and D. Selvi, “A distributed Kalman filter with event-triggered communication and guaranteed stability,” *Automatica*, vol. 93, pp. 75–82, 2018, doi: https://doi.org/10.1016/j.automatica.2018.03.005.
[172] F. S. Cattivelli and A. H. Sayed, “Diffusion Strategies for Distributed Kalman Filtering and Smoothing,” *IEEE Transactions on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, 2010, doi: http://dx.doi.org/10.1109/TAC.2010.2042987.
[173] V. Vahidpour, A. Rastegarnia, M. Latifi, A. Khalili, and S. Sanei, “Performance Analysis of Distributed Kalman Filtering with Partial Diffusion Over Noisy Network,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 3, pp. 1767–1782, 2020, doi: 10.1109/TAES.2019.2933961.
[174] G. Wang, N. Li, and Y. Zhang, “Diffusion distributed Kalmanfilter over sensor networks without exchanging raw measurements,” *Signal Processing*, vol. 132, pp. 1–7, 2017, doi: https://doi.org/10.1016/j.sigpro.2016.07.033.
[175] M. Kar and J. M. Moura, “Gossip and Distributed Kalman Filtering: Weak Consensus Under Weak Detectability,” *IEEE Trans. Signal Process.*, vol. 59, no. 4, pp. 1766–1784, 2011, doi: http://dx.doi.org/10.1109/TSP.2010.2100385.
[176] C. Wan, Y. Gao, X. R. Li, and E. Song, “Distributed Filtering Over Networks Using Greedy Gossip,” Cambridge, UK, Jul. 2018. doi: http://dx.doi.org/10.23919/ICIF.2018.8455499.
[177] H. Shin, S. He, and A. Tsourdos, “Sample greedy gossip distributed Kalman filter,” *Information Fusion*, vol. 64, pp. 259–269, 2020, doi: https://doi.org/10.1016/j.inffus.2020.08.001.
[178] M. A. Bakr and S. Lee, “Distributed Multisensor Data Fusion under Unknown Correlation and Data Inconsistency,” *Sensors*, vol. 17, no. 11, 2017, doi: http://dx.doi.org/10.3390/s17112472.
[179] M. Giraldo, F. Valencia, J. Espinosa, and J. D. L´opez, “Evaluation and Comparison Of Kalman-Filter-Based Distributed State Estimators For Large-Scale Systems,” in *Proceedings of the 13th IFAC Symposium on Large Scale Complex Systems: Theory and Applications*, China, 2013, pp. 188–193. doi: http://dx.doi.org/10.3182/20130708-3-CN-2036.00100.
[180] G. Ye and R. A. Fenner, “Kalman Filtering of Hydraulic Measurements for Burst Detection in Water Distribution Systems,” *J. Pipeline Syst. Eng. Pract.*, vol. 2, no. 1, pp. 14–22, 2011, doi: https://doi.org/10.1061/(ASCE)PS.1949-1204.0000070.
[181] A. M. Sadeghioon, N. Metje, D. Chapman, and C. Anthony, “Water pipeline failure detection using distributed relative pressure and temperature measurements and anomaly detection algorithms,” *Urban Water Journal*, vol. 15, no. 4, pp. 287–295, 2018, doi: https://doi.org/10.1080/1573062X.2018.1424213.

[182] M. Murnane and A. Ghazel, “A Closer Look at State of Charge (SOC) and State Of Health (SOH) Estimation Techniques for Batteries,” *analog devices*, pp. 1–8, 2017.
[183] Lady Ada, “Li-Ion & LiPoly Batteries,” *Adafruit learning system*, 2021. https://learn.adafruit.com/li-ion-and-lipoly-batteries/voltages (accessed May 20, 2021).
[184] EEMB Co., Ltd, “Lithium-ion Battery datasheet,” 2010. http://eemb.com (accessed May 20, 2021).
[185] R. A. Cody, P. Dey, and S. Narasimhan, “Linear Prediction for Leak Detection in Water Distribution Networks,” *J. Pipeline Syst. Eng. Pract.*, vol. 11, no. 1, pp. 1–16, 2020, doi: http://dx.doi.org/10.1061/(ASCE)PS.1949-1204.0000415.
[186] V. Nkemeni, F. Mieyeville, P. Tsafack, and J. Verdier, “Distributed Kalman Filter Investigation and Application to Leak Detection in WaterPipeline Monitoring Using Wireless Sensor Networks with Non-intrusive Sensors,” in *Proceedings of the Thirteenth International Conference on Sensor Technologies and Applications*, Nice, France, 2019, pp. 13–20.
[187] V. Nkemeni, F. Mieyeville, and P. Tsafack, “A distributed computing solution based on distributed Kalman filter for leak detection in WSN-based water pipeline monitoring,” *Sensors*, vol. 20, no. 18, pp. 1–38, 2020, doi: https://doi.org/10.3390/s20185204.
[188] Adafruit, “Adafruit Huzzah32 – ESP32 Feather Board,” May 10, 2017. https://learn.adafruit.com/adafruit-huzzah32-esp32-feather (accessed May 31, 2021).
[189] G. Owojaiye and Y. Sun, “Focal design issues affecting the deployment of wireless sensor networks for pipeline monitoring,” *Ad Hoc Networks*, vol. 11, no. 3, pp. 1237–1253, 2013, doi: https://doi.org/10.1016/j.adhoc.2012.09.006.
[190] TMRH20, “Optimized High Speed Driver for nRF24L01(+) 2.4GHz Wireless Transceiver,” 2014. http://tmrh20.github.io/RF24/ (accessed Feb. 18, 2020).
[191] TMRH20, “Network Layer for RF24 Radios,” 2015. https://tmrh20.github.io/RF24Network/ (accessed Feb. 18, 2020).
[192] A. Bounceur, “CupCarbon: A New Platform for Designing and Simulating Smart-City and IoT Wireless Sensor Networks (SCI-WSN),” NY, USA, 2016. doi: https://doi.org/10.1145/2896387.2900336.
[193] “CupCarbon User Guide Version U-One 5.1,” *CupCarbon*, 2021. www.cupcarbon.com (accessed Mar. 30, 2021).
[194] R. Muraleedharan, I. Demirkol, O. Yang, H. Ba, S. Ray, and W. Heinzelman, “Sleeping Techniques for Reducing Energy Dissipation,” in *The Art of Wireless Sensor Networks: Volume 1: Fundamentals*, H. M. Ammari, Ed. Berlin, Heidelberg: Springer, 2014, pp. 163–197. doi: 10.1007/978-3-642-40009-4_6.
[195] American Petroleum Institute, *API 1155: Evaluation Methodology for Software Based Leak Detection Systems*, 1st ed. American Petroleum Institute, 1995.
[196] G. Geiger, “State-of-the-Art in Leak Detection and Localization,” Hannover Messe, Hannover, Germany, Apr. 2006. [Online]. Available: https://www.pipeline-conference.com/abstracts/state-art-leak-detection-and-localisation
[197] A. Ayadi, O. Ghorbel, M. S. BenSalah, and M. Abid, “Spatio-temporal correlations for damages identification and localization in water pipeline systems based on WSNs,” *Comput. Netw.*, vol. 171, p. 107134, Apr. 2020, doi: http://dx.doi.org/10.1016/j.comnet.2020.107134.