IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

date of current version October 23, 2023.

Digital Object Identifier 10.1109/ACCESS.2022.0122113

A Self-Sustainable and Micro-Second Time Synchronised Multi-Node Wireless System for Aerodynamic Monitoring on Wind Turbines

TOMMASO POLONELLI[1], (Member, IEEE), AMIRHOSSEIN MOALLEMI[12], (Student Member, IEEE), WEIKANG KONG[1], HANNA MÜLLER[1], (Student Member, IEEE), JULIEN DEPARDAY[3], MICHELE MAGNO[1], (Senior Member, IEEE), and LUCA BENINI[12], (Fellow, IEEE)

[1]D-ITET Department, ETH Zürich, Zürich, 8092, Switzerland (e-mail: {topolonelli, amoallem, wekong, hanmuell, magnom, lbenini}@ethz.ch)
[2]DEI Department, University of Bologna, Bologna, 40132, Italy (e-mail: {amirhossein.moallem2, luca.benini}@unibo.it)
[3]IET Department, OST - Eastern Switzerland University of Applied Sciences, Rapperswil, 8640, Switzerland (e-mail: julien.deparday@ost.ch)

Corresponding author: T. Polonelli (e-mail: topolonelli@ethz.ch).

"This work was supported by the Swiss National Science Foundation (SNSF) Bridge Project "AeroSense" under Project 40B2-0_187087."

ABSTRACT Wind energy generation plays a vital role in transitioning from fossil fuel-based energy sources and in alleviating the impacts of global warming. However, global wind energy coverage still needs to rise, while requiring a significant step up in conversion efficiency: monitoring wind flow and operational parameters of wind turbines is an essential prerequisite for coverage and conversion efficiency optimization. This paper presents a low-power, self-sustainable, and time-synchronised system for aerodynamic and acoustic measurements on operating wind turbines. It includes 40 high-accuracy barometers, 10 microphones, 5 differential pressure sensors, and implements a coarse time synchronisation on top of a Bluetooth Low Energy 5.1 protocol tuned for long-range communications. Moreover, we field-assessed the node capability to collect precise and accurate aerodynamic data with a multi-node setup. Outdoor experimental tests revealed that the system can acquire heterogeneous data with a time synchronisation error below 100 µs and sustain a data rate of 600 kbps over 400 m with up to 5 sensor nodes, enough to fully instrument a wind turbine. The proposed method does not add any traffic overhead on the Bluetooth Low Energy 5.1 protocol, fully relying only on connection events and withstands transmission discontinuity often present in long range wireless communications.

INDEX TERMS Aerodynamic, Bluetooth, Time Synchronisation, Low Power, Sensors, IoT, Structural Health Monitoring, Wind Turbines, Energy Harvesting

I. INTRODUCTION

Wind energy has the potential to significantly scale up towards alleviating greenhouse gas emissions and help mitigating in climate crises and global warming [1]. To avert the worst impacts of climate change and preserve a livable planet, global temperature increase needs to be limited to 1.5 °C above pre-industrial levels, as called for in the Paris Agreement – emissions need to be reduced by 45% by 2030 and reach net zero by 2050 [2]. The urgency to tackle climate change has triggered a growing interest in wind energy [3]. To widely adopt wind energy as a leading power source, optimizing the design and operation of wind turbines to increase their efficiency while reducing negative environmental impacts is essential [4]. This requires a deeper understanding of the blade aerodynamics and an optimized environmental impact [5]. As for site selection, today, wind turbine installation is constrained due to significant noise emission and the subsequent complaints from nearby residents [6].

It is essential to gather pressure gradient and audio data from operational wind turbines to gain a deeper understanding of aerodynamics and acoustic phenomena [7]. Given the relative novelty of this research field, there is a clear demand for advanced and continuous

monitoring solutions to facilitate data collection, particularly for generating publicly available datasets [8]. Supervisory Control and Data Acquisition (SCADA) systems have been extensively employed in various industrial control and Industrial Internet of Things (IIoT) applications [9], but their utilization in the wind turbine industry is limited. Currently, these systems mainly acquire data from static components rather than from the rotating blades [10]. This limitation arises due to the considerable challenges associated with acquiring aerodynamic and acoustic data from the blades of hundreds-of-meters-tall turbines, which often involves significant effort and costs in conducting measurement campaigns [8], [10].

Recent efforts have been made to address this issue and obtain data from moving parts. For example, in [11], researchers manufactured custom blades equipped with embedded Pitot tubes and acoustic sensors [11], [12] to measure the airstream and turbulence impacting the rotor blade. However, a growing trend is to replace such expensive and wired measurement systems with scalable and low-power IIoT sensors in the next generation of monitoring systems [8], characterized by reduced installation costs. In this direction, the Aerosense system [13] has recently proved to support in-situ measurements directly on a wind turbine blade, exploiting a flexible and ultra-thin (<4 mm) wireless sensor node. Aerosense is a low-power, self-sustainable sensor node for aerodynamic and acoustic measurements on wind turbines. It includes 40 high-accuracy barometers, 10 microphones, and 5 differential pressure sensors [14]. The wireless transmitter is based on Bluetooth Low Energy 5.1 (BLE) tuned for long-range while maintaining high per-bit energy efficiency (80 nJ) [14], supporting a data rate of 850 kbps over 438 m [8]. Previously cited works show the possibility of acquiring aerodynamic data directly from an operating wind turbine [8], [9], [11], [13], [14]; however, they only address a single node installation, without considering the relevant challenges of monitoring an entire wind farm composed of tens of electric generator and hundreds of blades. To enable the multi-node installation of several sensor units, precise time synchronization needs to be present to enable data alignment in post-processing, which is a general challenge for low-power wireless devices.

For the scope of this paper, the Aerosese system [14] has been used as a reference point. We focus on enhancements required or support a multi-node installation. We leverage the Aerosense framework to assure back-compatibility, enriching its original features with precise time synchronization down to tens of microseconds, supporting a multi-node connection with a tree topology and a remote server, and assessing the data traffic in different configurations.

The objectives of this paper are: (i) Feasibility analysis of using multi-node Internet of Things (IoT) networks to monitor an entire wind turbine directly with field measurements on operating electric generators: this is a study that, to the best of our knowledge, has never been conducted before; (ii) A full-stack description of a distributed measurement system which is low power, self-sustainable, and scalable. We describe both hardware and software and give a detailed and system-level overview; (iii) The study, analysis and implementation of a precise and robust time synchronization technique over a BLE wireless link with minimal traffic overhead, exploiting only intrinsic BLE packet exchanges, and reaching an average synchronization error of 40 µs and a maximum of 100 µs; (iv) A clock drift monitoring with error compensation to correct the internal Real Time Clock (RTC) drift due to variations in temperature and aging; (v) An experimental analysis of the bandwidth limitation of the wireless link was conducted from one to five sensor nodes. Collected experimental results show a maximum throughput of 2 Mbps with a single-transmitting node. For a multi-node deployment, a bitrate of 600 kbps was empirically determined to guarantee transmission robustness over 400 m with up to 5 sensor nodes connected to a single base station. Due to the high amount of generated data from each sensor, which can reach up to 4.2 Mbps, and the bandwidth limitation over long-range wireless links, the transmission time can exceed by 7× the sensors acquisition period, up to 35× with 5 sensor nodes transmitting in sequence. These collective bandwidth limitations thereby enforce the necessity of an accurate time synchronization for re-aligning the collected data in post-processing. (vi) A multi-node installation on an operating wind turbine, Aventa AV-7 turbine [15], equipped with two time-synchronized sensor nodes on the same blade. The test, conducted over one month, demonstrates with experimental results the effectiveness of our method of keeping the sensor operational for turbine monitoring with a synchronization error not exceeding 100 µs. Even in the presence of packet loss and miss-communications, the maximum time misalignment between two sensor nodes remains below 160 µs.

Section II presents a literature review focused on sensor nodes for wind turbines, with specific emphasis on Bluetooth time synchronization and low-power sensors. Section III describes the application scenario and its specific requirements, derived from the goal of performing wind turbine monitoring through aerodynamic analysis. In Section IV, we delve into the details of our low-power sensor node, named Aerosense, describing all its components, from the hardware and the energy sources to the firmware and wireless transmission sub-system. Moreover, an overview of the whole Aerosense system is provided. Section V describes the wireless time synchronization protocol, supporting a multi-node installation and micro-second precision. Unit tests and performance analysis are reported in Section VI. Sec-

tions VII and VIII report results from a real installation on an operating wind turbine. Lastly, Section IX concludes the paper with an overview of future applications and research directions.

II. RELATED WORKS

Monitoring the structural health of civil structures, commonly referred to as Structural Health Monitoring (SHM), has garnered significant research attention in recent years [8]. Focusing on wind turbines, previous studies have explored the integration of various sensors, such as pressure sensors [16] and microphones [8], to demonstrate their importance in real-time control and intelligent maintenance. Additionally, researchers have investigated the monitoring of blade airflow using distributed systems [17].

While earlier analyses have shown the potential of utilizing cost-effective Micro-Electro-Mechanical System (MEMS) sensors for SHM applications [18], only a few works in academia or industry have presented MEMS multi-sensor arrays specifically designed for wind turbines [14]. Previous studies have examined arrays of barometric MEMS sensors in various application scenarios, such as mounting them on aerodynamic surfaces like airplane wings [19] and cars [20]. However, these works primarily focus on acquiring aerodynamic and aeroacoustic measurements without addressing the challenges of wireless communication, power consumption, and the requirements for continuous and long-term monitoring of wind turbines, particularly when employing a multi-sensor setup [21].

Within the Aerosense work [22], authors successfully demonstrate the effective application of MEMS sensors for measuring sound levels and pressure distribution on wind turbine blades, deploying a self-sustaining sensor node wirelessly connected to a base station, able to function 24/7 on an operating wind turbine. Further, Wondra et al. [23] propose a flexible solution for SHM of wind turbine towers using only MEMS accelerometers in a battery-powered and wireless system. However, despite deploying the system on a wind turbine, they do not provide information on power consumption or estimated battery lifetime. Another comparable solution is presented by Lu et al. [24], where the health status of a planetary gearbox is estimated through vibration analysis. This work also demonstrates the practical implementation of a vibration-based energy harvester utilizing a 0.9 mW piezoelectric vibration energy harvester to fulfill the energy demands. Nonetheless, the sensor node in this solution has a low sampling frequency of 50 Hz and limited coverage of 6 meters.

While a few examples of wireless solutions installed on wind turbines can be found in the literature [25], previous efforts have primarily focused on vibration measurements on the wind turbine tower rather than acquiring aerodynamic data directly from the blades. In the former case, the system must process and transmit data in the range of 5 kbps [24], while aerodynamic analysis typically requires a much higher throughput of over 1 Mbps [8], also reaching up to 4.2 Mbps when microphones are employed [14]. The additional data collected by barometers and microphones in multi-sensor arrays is crucial for wind turbine modeling but presents significant challenges in designing energy-efficient and long-lasting IoT devices. More importantly, the high requirements in the wireless bandwidth pose a challenge for system scalability, as the maximum number of sensing elements is bounded by the effective wireless channel bandwidth [26]. These works feature low transmission data rate, and they are tested and designed to work with a single agent rather than collaborating, gathering structural information at the system level from a heterogeneous cluster of sensor nodes.

In the domain of wireless measurement systems for wind farms, a comprehensive technical comparison has been presented in [27]. For a real-life deployment scenario, we focus on Aerosense, a state-of-the-art wireless system explicitly designed for monitoring and analyzing the performance and structural health of wind turbines [13]. While the Aerosense system has been proved in the field by exploiting a long-range BLE wireless link, the presented results only focus on a single-node installation without investigating the possibility of extending to multi-node use cases.

Two main requisites are necessary to deploy a multi-node setup, namely, network scalability in terms of collective bandwidth requirements and precise time synchronization to align information collected by several sensor nodes [28]. In the field of aerodynamic wind turbine monitoring, we additionally need to comply with the limited processing capabilities of the embedded MicroController Unit (MCU) and the power consumption, which needs to be minimized to reach long-term measurement campaigns [29]. Thus, for this paper's scope, the BLE long-range wireless link featured by Aerosense is enhanced to support a multi-sensor deployment with time-synchronized measurements [30].

A. BLE TIME SYNCHRONIZATION

There are two major types of BLE time synchronization methods: connectionless and connection-based [31]. In connectionless BLE time synchronization methods, the master device broadcasts beacons to perform the synchronization. Authors in [32] present a BLE time synchronization solution in an application that requires synchronization between the transmitter and the receiver. In the solution, named CheepSync, the transmitter broadcasts a single advertisement packet containing the transmitter's timestamp. The timestamp corresponds to the time at which the packet is transmitted. Upon receiving the timestamp, the receiver records the local timestamp and calculates the time offset between the

transmitter and the receiver. CheepSync makes continuous skew adjustments over a measurement window to handle clock drift. By doing so, it can maintain a synchronization error of 10 µs with a re-synchronization every 100 ms.

To handle the challenges in connectionless BLE time synchronization, such as single-channel scan, random BLE delay, and low-error timestamping, authors in [33] propose BlueSync. To avoid keeping broadcasting and listening for the beacons and therefore save energy, the master only broadcasts beacons in the synchronization time slot, which is just a few seconds, and there is no radio activity in the following 10-minute synchronous task. The clock drift is estimated with the broadcasted timestamps and is used to keep the devices synchronized. The underlying assumption is that the clocks drift apart with the same slopes in the synchronous task as in the synchronization time slot. To achieve a better synchronization than the onboard 32.768 kHz RTC can offer, a 16 MHz timer clock is used to compensate for the clock resolution and finally achieves an average synchronization error of 320 ns per 60 s with a timer clock period of 62.5 ns. However, in [34], the authors point out that in a practical deployment, the time-synchronization delay caused by the packet transmission process can strongly affect the accuracy and precision of the time synchronization.

Although the above-mentioned methods have their merits, they are not suitable for the application scenario of this paper for two major reasons. One is the necessity of maintaining a connection to acquire connection statistics for analytical purposes and reliably and securely transmit some data while keeping the devices synchronized. It is also possible to configure two GAP roles (Broadcaster and Central for the base station, Observer and Peripheral for the sensor node), but that comes with an increased complexity of configuring one more GAP role (Broadcaster for the base station and Observer for the sensor node) that unnecessarily consumes memory resources when the connection can already be utilized to perform synchronization. The second reason is that the clock drift estimation technique, as presented in [33], is difficult to apply in our scenario, as sensor nodes installed on an operating wind turbine suffer from substantial temperature changes, which can strongly impact clock drifts.

For connection-based synchronization, there are two major types of timestamping methods based on the literature. The first method utilizes external circuits to monitor the current consumption changes caused by BLE events for precise timestamping. Authors in [35] propose to monitor the current profile of a connection event and generate a hardware Interrupt ReQuest (IRQ) when the output of the Customer Service Management (CSM) exceeds a threshold. The generated IRQs at the master and the slave are delayed with a standard deviation of 0.9 µs. In [36], different options for using current measurement for BLE time synchronization are explored. They look into the current profiles of the master and the slave during connection establishment, connection events, and disconnection. The authors conclude that "many of these events occur almost at the same time and can be used for precise time synchronization" [36]. Current-measurement-based timestamping methods are promising but require external circuits, increasing the system's complexity, cost, and power consumption.

BLE time synchronization based on radio events is challenging when it only operates at the application layer due to the non-deterministic delay in transmissions. Authors in [37] measure the time difference between the connection events on the master and slave devices and achieve an accuracy of time synchronization within ±750 µs. Authors in [38] record the timestamps when a packet is transmitted and received at the master's and the slave's sides to calculate the clock offset and reach a Root Mean Square (RMS) timing error of 6.6 µs with a 64 MHz clock to timestamp the samples. The achieved synchronization errors in these methods are not low either in terms of absolute value or relative to the timestamping clock period. An RMS error as low as 20 µs was achieved in [39] with a 32 kHz RTC assuming that the operation timings of the sensor node's (SN) and the base station's (BS) feature a time difference with a small standard deviation. However, according to the protocol description in [39], the underlying assumption is that the timestamps TS_{SN} from the sensor node and TS_{BS} from the base station are collected in the same connection event, which does not hold when the data packet from the sensor node re-transmits, a common situation in real deployments. In this case, the protocol will mistakenly assume TS_{SN} and TS_{BS} are collected in the same connection event when they are not, resulting in the misalignment between the timestamps collected at both sides. In the case of re-transmission, the calculated offset deviates from the case where no re-transmission happens by a multiple of Connection Interval (CI), depending on the number of re-transmissions.

The aforementioned works show the possibility of enabling microsecond time synchronization over the standard BLE protocol. Moreover, they show the possibility of exploiting the connection events to decrease the synchronization transmission overhead, keeping the traffic and power consumption under control. Despite their potentiality, these works target specific application scenarios or are tested in controlled environments, not supporting common problems in real field deployments, such as packet loss and concurrent re-transmissions. This is where our work is focused on designing and implementing a reliable, low-power, and efficient synchronization protocol with micro-second accuracy in the field of wind turbine monitoring.

III. SENSING REQUIREMENTS FOR WIND TURBINE AERODYNAMICS MONITORING

To understand the technical design choices and the Aerosense system setup, a basic overview of wind turbine aerodynamics has to be provided.

The airfoil is defined as the cross-section of a wind turbine blade. For simplicity, in this paper, we consider a symmetrically shaped airfoil, where the chord, the distance between Leading Edge (LE) and Trailing Edge (TE), is shown in Figure 1. One of the key variables of a blade is the Angle of Attack (AoA), defined as the angle between the airflow direction and the blade chord. With an AoA in the $0 - 20°$ range, the absolute pressure on the wind-facing side (described as pressure side in Figure 1) increases while it decreases at the opposite edge on the suction side. In this condition, the pressure distribution surrounding the blade creates the lift force, pulling the blade towards the suction side. When the AoA increases too much, the fluid flow that normally moves over the suction surface generates a flow separation near the TE, causing turbulence, audible noise and dramatically decreasing the lift effect. On the other side, an ideal symmetric airfoil does not generate any lift for AoA equal to zero. Hence measuring the pressure all around the blade surface is a fundamental prerequisite for analyzing the dynamic behavior of a wind turbine during its normal operation in the field, both for increasing its performance with accurate real-time control and for accurate model generation. In the case of large wind turbines, and multi-MW generators, a blade can be longer than 100 m; thus, a measurement system able to acquire operational data from different points is a primary necessity to model its dynamic aerodynamic behavior.

For the scope of this paper, the minimum pressure accuracy was determined for wind turbines of two different sizes, Aventa AV-7 6 kW wind turbine [15] and DTU 10MW wind turbine [40], in the range between 5 Pa and 40 Pa with a dynamic working range above 1 kPa. Moreover, the needed minimum barometric accuracy was defined to be at most 1% of the dynamic pressure range at average wind speeds [8]. Additionally, the chosen sensor must have a low drift in temperature [16] and a sampling rate equivalent to 100 times per rotation [8], [16]. This amounts to approximately 70 samples per second for the Aventa AV-7 turbine [15] and 17 sps for the DTU 10 MW generator [40]. In [41], the authors developed and demonstrated an identification method for transitional, and turbulence flows (flow separation in Figure 1) relying on audio data. In a more recent work [14], a similar approach was replicated using an array of 10 MEMS low-power microphones and a bandwidth up to 8 kHz. These works also provided the maximum estimated acoustic pressure level at 500 Pa, corresponding to a Sound Pressure Level (SPL) above 140 dB SPL.

IV. SYSTEM ARCHITECTURE

This section describes a full-stack description of the system named Aerosense [13]. It is primarily based on three main components: a sensor node to acquire and handle the data from different locations of the blade (Section IV-A), a base station to forward the data from different nodes via the gateway to the cloud (Section IV-B), and the cloud infrastructure to record and process the acquired data with a digital twin. This section starts by giving a general overview of the system to acquire heterogeneous data from various sensors directly from wind turbine blades, successively transmitting and processing it in a digital twin. Further, sensors and wireless System on Chip (SoC) are described as responsible for collecting and transmitting data from the physical world to the base station. Additionally, the base station system and its connectivity to the cloud are described. Figure 1 presents a high-level system overview.

The Aerosense [13] system was designed [14], tested, validated in the field [14], and calibrated [16] in previous works. These previous studies also characterize the system power consumption, the photovoltaic energy harvester with a flexible 74 mm x 146 mm solar panel MPT3.6-150 from PowerFilmSolar[1], and the battery with a 32 Wh (8.7 Ah) lithium element. However, the system has been limited so far by installing only a single sensor node, which poses a practical restriction to monitoring wind farms or large multi-MW generators. In this paper, we propose a brief description of the Aerosense system already tested on an operating wind turbine [22] as a base to extend its functionalities with a multi-node and time-synchronized setup.

A. AEROSENSE SENSOR NODE

The main Aerosense device is a self-sustainable and long-range Bluetooth wireless sensor node for heterogeneous measurement directly on the surface of an operating wind turbine blade. Each sensor node is directly controlled by its associated base station that initiates and configures the sensor node to collect data and for wireless transmissions. Particularly, upon powering up the system, the base station starts by sending a chain of commands to the sensor nodes via the BLE transceiver to configure the data acquisition process, such as sensor sampling rate, power state, enabled sensing elements, and etc. Once the configuration is done, each sensor node can start collecting data from the blade for a predetermined period of time from different locations of the blade, as shown in Figure 1. The data is first recorded in the internal non-volatile memory during the acquisition process, in which each sample is associated with a timestamp retrieved from the local clock. After the data collection phase is completed, the base station

[1] www.powerfilmsolar.com

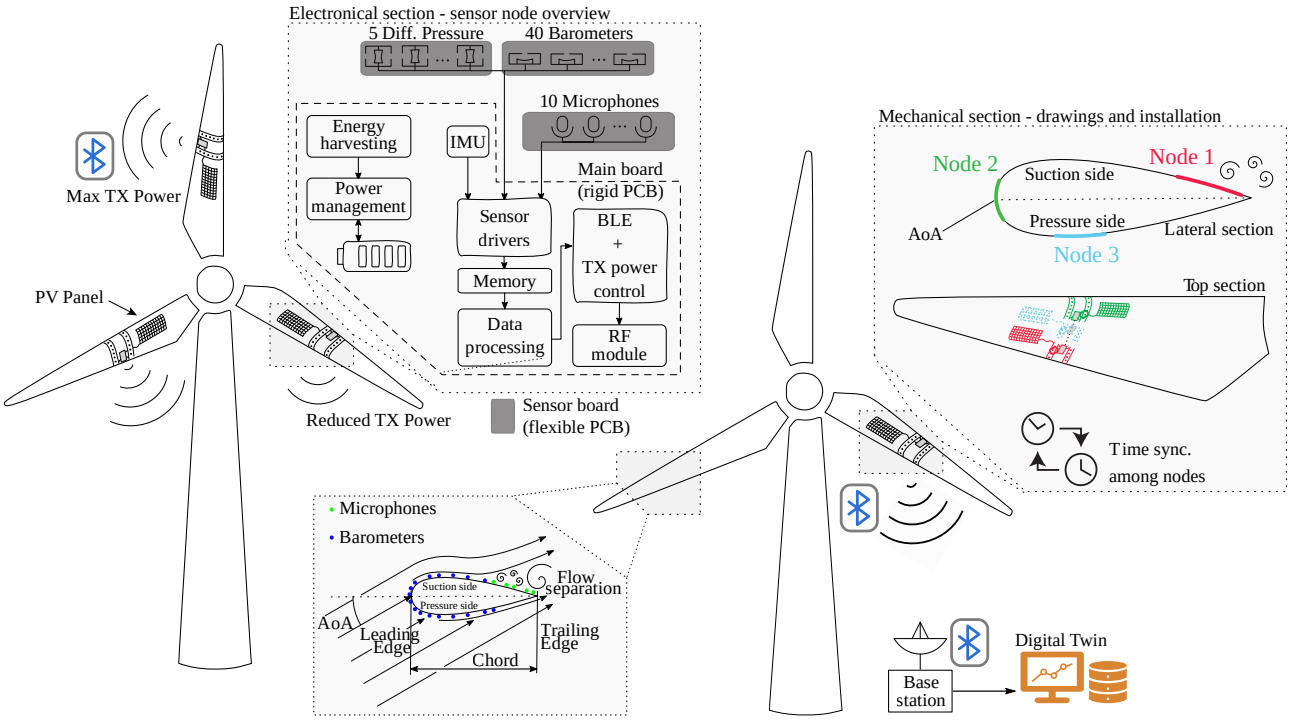


FIGURE 1: A high-level overview of the proposed monitoring system and the Aerosense sensor board. Aerosense supports multi-node installation over multiple blades and wind turbines. Details of the electronic, mechanical installation, and aerodynamic main definitions are provided.

transmits the read commands for data transmission from the cloud. The recorded data are then transmitted to the base station via BLE and subsequently to the cloud storage via an internet connection. This procedure is designed to decouple the sensor sampling from the wireless transmissions. Thus, each onboard sensor can be accurately sampled, removing the real-time communication constraints, including bandwidth limitation and packet loss. In a scaled system, multiple Aerosense nodes can be deployed to monitor the wind turbine, where the BLE synchronization mechanism is established before the data acquisition to ensure a uniform and accurate data acquisition process.

1) MEMS Sensors

The Aerosense node features various sensors that collect heterogeneous data. These sensors are designed to acquire specific aerodynamic and acoustic parameters, namely pressure, temperature, audio, and vibrations. Table 1 summarizes the selected commercial MEMS sensors in terms of power consumption, height, and generated bitrate. To characterize the pressure distribution along the blade, measuring the pressure across the airfoil is necessary. To this end, a set of 40 absolute pressure sensors have been positioned at predetermined intervals along the airfoil, Figure 1 showcases the placement of barometers as blue solid dots when the chord length is below 1 m and a set of multiple sensor clusters to cover longer chord lengths (namely Node 1-3 in Figure 1). Considering the sensor's power consumption and thickness, ST LPS27HHW is selected as the absolute barometer. Further, this sensor benefits from an integrated temperature sensor, allowing on-site calibration to avoid temperature drift and compensate for aging over time. ST LPS27HHW is a water-resistant, ultra-compact absolute barometer sensor that can measure from 260 to 1260 hPa absolute pressure range yielding the aerodynamic requirements of a wind turbine monitoring system. A previous study demonstrated the possibility of achieving the required 5 Pa absolute accuracy via calibration [16].

In addition to the absolute pressure sensor, Aerosense also provides the possibility of having true differential sensing via 5 differential Pewatron 52 pressure sensors. They are used to estimate the local wind conditions at the blade [16]. It offers a compact sensing unit with only 5 mm with an integrated multi-order compensation algorithm for correcting offset, sensitivity, and thermal errors. Finally, a sampling rate of 1.2 kHz with a resolution of 16-bit is chosen for the Aerosense implementation.

To record and monitor the audio spectrum of the flow separation effect on the wind, the Vesper VM2020 MEMS microphone is chosen. Microphones' high bandwidth tends to generate a large number of data making it non-trivial for a tiny device to manage it. Subsequently, among the commercial microphones, the VM2020 is

TABLE 1: Aerosense set of sensors. For each sensor the generated bitrate is calculated based on the selected sampling rate.

Model/Quantity	Sampling rate [Hz]	Power [μW]	R/A$^{\diamond}$ accuracy	Height [mm]	BW† [kbps]
BMX160 / 1	100	83	46 mg$^{\Delta}$	0.95	9.6
LPS27HHW / 40	100	7.2	$\pm$2.5/$\pm$100	1.7	96
Pewatron 52 / 5	1.2 k	7.3	$\pm$10/$\pm$50	5	294
Model	**Sampling rate [kHz]**	**Power [μW]**	**AOP* [dB]**	**Height [mm]**	**BW† [Mbps]**
VM2020 / 10	16	446	152	1.3	3.8

* Acoustic Overload Point, $^{\Delta}$ Full scale $\pm$16 g
$^{\diamond}$ Relative/Absolute accuracy, † Generated bitrate

selected with 6 kHz bandwidth and a dynamic range of 140 dB SPL to respect the aerodynamic requirement explained in Section III.

Vibration-based sensors are widely used in SHM complex systems to study the integrity of the structures. Further, a correlation between pressure and audio with the placement of the blade is vital for the aerodynamic and acoustic analysis of wind turbines [14], [16] to correctly remove the altitude variation generated by the intrinsic periodic wind turbine oscillation, from the collected data. Thus, a 9-axis Inertial Measurement Unit (IMU), namely, Bosch BMX160, is placed in the Aerosense design as the final piece of the sensing unit. BMX160 consists of a 3-axis accelerometer, 3-axis gyroscope, and 3-axis magnetometer, providing high accuracy with low current consumption, 46 µA in active mode and 4 µA in idle mode.

2) Wireless System on Chip (SoC)

The wireless SoC chosen for the Aerosense node is the CC2652P by Texas Instrument. It embeds a 48 MHz ARM Cortex-M4 processor, a 2.4 GHz wireless interface, and a wide range of peripherals to link external sensors. Mainly, the CC2652P offers a low power consumption for sensor readout, integrating a Sensor Controller (SC) unit. This provides the capability of deploying one line of SPI or I2C at 30 µA in addition to the rest of the peripheral unit. Moreover, CC2652P benefits from 352 kB of in-system programmable flash and 88 kB of SRAM. Finally, the BLE wireless interface provides a long-range transmission respecting the requirements of large-scale structures like wind turbines with a data transfer rate of up to 2 Mbps [8]. A combination of low energy and long-range transmission range (up to 400 m) makes BLE a suitable candidate for wireless sensing systems such as wind turbines.

Table 1 shows that the sensor node needs to support a large data bandwidth; only the ten microphones lead to a total of 3.8 Mbps. Considering all the sensors together, the bandwidth amounts to a total of 4.2 Mbps ($\mathrm{Tot_{BW}} = \mathrm{BW_{BMX160}} + \mathrm{BW_{LPS27HHW}} + \mathrm{BW_{Pewatron52}} + \mathrm{BW_{VM2020}}$). Since the BW of the system is twice as BLE BW, continuous-stream transmission of data to the base station would not be possible. Thus, time windows for data collections are defined by the base station, and then the sensor node collects time-stamped data and stores it on a 512 MB flash memory, namely Kioxia FLASH TC58CYG2S0HRAIJ, is selected as a local non-volatile buffer. Finally, the time-stamped data is transmitted to the base station according to a predefined schedule. The base station will use the time stamps to align in time data coming from multiple sensors. Thus accurate synchronization is essential to ensure precise alignment.

B. BASE STATION

The base station serves as a key component in the Aerosense system, acting as an intermediary between the sensor nodes and the cloud. It comprises a BLE transceiver and a local computing unit that collects data from the local sensor nodes and forwards it to the cloud storage for digital twin analyses. The base station's role in the Aerosense system is crucial for establishing a reliable and efficient data collection process, ensuring that all the sensor nodes are active and properly working. Indeed, the base station is in charge of transmitting commands to the sensor node to control the system's flow. Moreover, the base station contributes two other pieces of information to the data collected by the wireless sensor nodes, such as atmospheric pressure and air temperature collected at the ground used as a reference for the whole system.

The computing unit that controls the base station is the Raspberry Pi 4 with a 64-bit ARM-Cortex A72 running at 1.5 GHz. It supports a Linux distribution, allowing Python applications to properly handle the data from the BLE receiver and send them to the cloud over the MQTT protocol and cellular wireless connection.

V. TIME SYNCHRONIZATION OVER BLE FOR A MULTI-NODE SETUP

From the literature and the Bluetooth protocol specifications, two different time synchronization methods over BLE can be considered: connectionless and connection-based methods. In this section, the base station is named Central device, and the sensor node is called Peripheral device as in the Aerosense context. This paper presents a two-step BLE connection-based time synchronization protocol with minimal overhead regarding the processing load of the MCU, memory, power consumption, and packet traffic on the BLE channel. Thus, the single connection GAP role used to send commands to the sensor nodes and subsequently download the data to the base station is also exploited for synchronization. Indeed, to avoid keeping broadcasting and listening for the connectionless beacons and therefore save energy, the master only broadcasts the BLE connection events (Figure 2) in specific time slots, which is just a few microseconds. In practice, it would also be possible to configure two GAP roles (Broadcaster and Central for

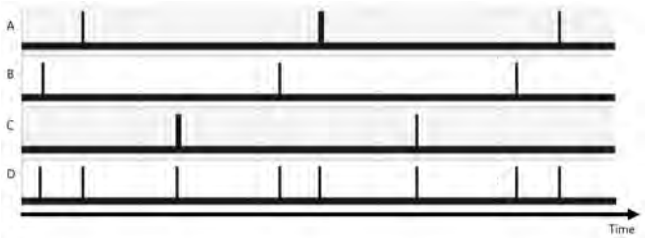


FIGURE 2: Transmission signals from one base station and three sensor nodes. Signal A for node 0, B for node 1, and C for node 2 of the tx signal, where signal D is the base station tx signal.

the base station, Observer and Peripheral for the sensor node), but that comes with increased complexity of configuring one more GAP role (Broadcaster for the base station and Observer for the sensor node) that unnecessarily consumes MCU and battery resources when the single role connection can already be utilized to perform synchronization. Thus, the connectionless approach is not further investigated. Moreover, this section also covers the link layer and connection event protocols to support a reliable and flexible synchronization strategy also in case of packet losses, Peripheral ghosting and re-transmissions. For example, counteracting for the wrong alignment issue discussed in Section II using connection event counters. Therefore, we propose a two-step time synchronization protocol that features: (i) a coarse synchronization step which aims for a coarse clock offset compensation over BLE (ii) a fine synchronization step in which the coarse offset is used to help align the fine timestamps collected at each side to calculate the fine offset. In the rest of the section, T refers to the global time, and t refers to the local RTC time on each device.

A. COARSE SYNCHRONIZATION

To correctly understand the basis of the proposed methodology, the connection event sequence featured by the BLE protocol needs to be defined. For this reason, we provide here a specific example aligned with our case study. In Figure 2, one base station is connected with three sensor nodes, where the transmission signal events are sampled with the logic analyzer of a DIGILENT Analog Discovery 2 at 1 MHz. At the base station's side, there are three pulses in the time span of one CI (in this figure CI = 100ms), each corresponding to the start of each connection event with each sensor node.

At the base station's side, the time difference between the start of two consecutive transmissions for one sensor node is always approximately equal to a multiple of CI, with a negligible deviation. Besides, in one connection event, the sensor node transmissions' end and start are approximately spaced by 90 µs.

To build the coarse synchronization, the connection sequence needs to be determined to enable a fast and lightweight synchronization procedure. In the implementation, we decided on a static sequence, where the sensor node with connection index 0 is always regarded as the first in the connection list. During the connection sequence, the base station first registers the connection event with the first sensor node in the connection list. At the same time, the RTC counter register (i.e., RTC timestamp) is read with direct memory access (this operation is referred to as "read the RTC timestamp" in the rest of the text), and the callback is registered with the next sensor node in the connection list. The list is bounded by the MCU memory size, and for the scope of this paper, it was limited to 5. The collected timestamps are mapped into the range $[ts_0, ts_0 + CI]$, where ts_0 is the timestamp for the first sensor node in the connection list, with the function as shown in Equation (1), where N is the number of connected sensor nodes.

$$\tilde{t_i} = f(t_i) = t_i - \lfloor \frac{t_i - t_0}{CI} \rfloor \cdot CI, i = 1, ..., N - 1 \ . \quad (1)$$

Ascending sorting is then performed on the mapped timestamps, and the subscripts of the sorted timestamps reflect the connection sequence.

After this step, the coarse synchronization begins, where Figure 3 displays the flow chart of each step. The base station starts the procedure by sending the "startCoarseSync" command message to all the connected sensor nodes. Upon receiving the command, each sensor node registers the connection event callback, in which the RTC timestamp t_s^{temp} is recorded with a near-zero delay. Note that the t_s^{temp} is updated at every connection event until the sensor node is unregistered.

After the commands are sent to all the sensor nodes, the base station queues an indication to the first sensor node in the connection sequence list. In the callback for connection event i, the RTC timestamp t'_b is saved, and the event is unregistered. At the sensor node's side, the indication can be successfully received in the same connection event (i.e., connection event i) at the earliest, but this condition does not always apply. After receiving the indication, the latest recorded timestamp t_s^{temp} is saved as the coarse reference timestamp t'_s and a dedicated timer is scheduled to be invoked in k millisecond, where k ms is a sufficiently long time after which the current connection event (connection event i at the earliest) has finished. When performing the coarse synchronization, the application data transmission is minimized, and the connection event duration is usually small; therefore, setting k as CI/2 is a practicable option. In the timer callback, the indication confirmation is queued to the BLE stack for transmission. When the current connection event i has finished, the confirmation can be successfully received by the base station in the next connection event (i.e., connection event i+1) at the earliest. When the base station receives the indication confirmation, the RTC timestamp is read and saved as t_b^{cfm}. By doing so, the time difference between the two

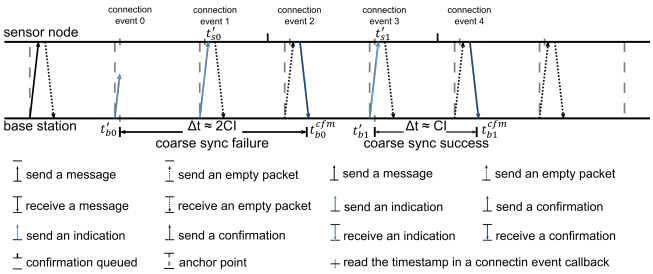


FIGURE 3: A diagram showing the coarse synchronization sequence from left to right over time. Messages name, sequence, and time references are reported.

saved timestamps at the base station $\Delta t = t_b^{cfm} - t'_b$ satisfies Equation (2), where R is the ratio of Δt to CI.

$$R = [\frac{\Delta t}{CI}] \geq 1 \ . \tag{2}$$

If R is exactly 1, the indication is successfully transmitted and received in the same connection event as the timestamp t'_b is saved, and then the confirmation is exactly transmitted and received in the next connection event. t'_b and t'_s are considered the coarse reference timestamps for the base station and the sensor node. The coarse clock offset "offset′" can be calculated as in Equation (3).

$$\text{offset}' = t'_s - t'_b \ , \tag{3}$$

In the case of R different than 1 , three possibilities have to be considered: (i) The connection event callback registration and the queuing of the indication are performed shortly before the start of a connection event. The registration is finished before the connection event, but the indication is successfully queued to the BLE stack after the connection event has started. (ii) The indication is lost and then re-transmitted. (iii) The confirmation is lost and then re-transmitted; in this case, the condition $R > 1$ applies. In case (i) and (ii), the saved timestamps t'_b and t'_s cannot be directly used for the calculation of coarse clock offset. In case (iii), the timestamps can actually be used, but there is no access to the link layer acknowledgment, so whenever $R > 1$, the coarse synchronization is considered a failure and, thus, discarded. In the case of $R > 1$, another coarse synchronization is performed until $R = 1$. The possibility of events such as case (iii) validates the technical motivation to not directly calculate the desired coarse clock offset with Δt, t'_b, and t'_s since it would result is an incorrect estimation. However, in cases (i) and (ii), the coarse clock offset can still be inferred according to Equation (4).

$$\text{offset}' = t'_s - (t'_b + (R-1)\cdot CI) \ , \tag{4}$$

After the coarse synchronization for one sensor node succeeds, the base station proceeds to the next one in the connection sequence list. It is done in this sequence to minimize the waiting time between the end of one coarse synchronization loop and the start of the next. From this moment on, each sensor node can internally calculate a network-aligned timestamp ($\text{timestamp}_{\text{sync}}$) simply applying Equation (5).

$$\text{timestamp}_{\text{sync}} = \text{offset}' + \text{RTC}_{\text{timestamp}} \ . \tag{5}$$

However, despite $\text{timestamp}_{\text{sync}}$ offering a robust time reference across the network, it cannot cover non-idealities such as internal clock drift, CI period variability and temperature dependency. Issues that need to be separately addressed with a dedicated method.

B. FINE SYNCHRONIZATION

At the base station's side, after the coarse synchronizations with all the sensor nodes have finished, the base station sends "startFineSync" command to them. In the meantime, the base station maps the transmission signal to a physical interrupt line and performs falling edge detection. RTC channel 1 is configured to capture mode to latch the current timestamp upon an edge detection event. Furthermore, the edge detection also triggers a function hooked to RTC combined interrupt to read out the latched timestamp in time. RTC channel 1 and the transmission signal mapping are disabled when N_b timestamps ($\mathbf{t}_b$) have been collected.

At the sensor node's side, the connection event callback registered in the coarse synchronization step is unregistered upon receiving the "startFineSync" command. The sensor node maps the transmission signal to a physical interrupt line for edge detection. RTC channel 1 is disabled when N_s timestamps have been collected, but the mapping continues.

N_b and N_s should satisfy Equation (6), where M is the number of devices connected to the base station.

$$N_b = M \cdot N_s \ . \tag{6}$$

After N_s timestamps have been collected, the sensor node transmits these timestamps $\mathbf{t}_s$, together with the previously collected coarse reference timestamp t'_s to the base station.

C. CLOCK OFFSET CALCULATION

After receiving the timestamps from all the sensor nodes, the base station starts the clock offset calculation. The coarse offset is calculated for each sensor node as in Equation (3). Then a timestamp in $\mathbf{t}_s$ is picked as t_s^{ref}, and the matched timestamp at the base station's side is estimated according to Equation (7).

$$\hat{t}_b^{ref} = t_s^{ref} - \text{offset}' \ , \tag{7}$$

$$\begin{cases} j = \underset{i}{\text{argmin}} |\mathbf{t}_b(i) - \hat{t}_b^{ref}| \ , \\ |\mathbf{t}_b(j) - \hat{t}_b^{ref}| < \Theta \ . \end{cases} \tag{8}$$

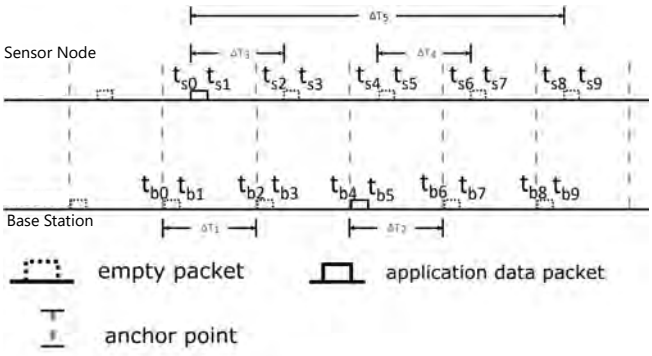


FIGURE 4: A diagram showing the drift monitoring sequence from left to right over time. Messages name, sequence, and time references are reported. ΔT is not always a multiple of CI.

If there exists a "j" that satisfies Equation (8), then it is considered that the base station fine reference timestamp $t_b^{ref} = t_b^j$ is collected in the same connection event as t_s^{ref} (t_b^{ref} aligned with t_s^{ref}). If such a matched timestamp cannot be found, another timestamp from $\mathbf{t_s}$ is chosen as t_s^{ref}. If none of $\mathbf{t_s}$ has a matched timestamp in $\mathbf{t_b}$, the fine synchronization procedure is repeated. The clock offset is calculated as in Equation (9).

$$\text{offset} = t_s^{ref} - t_b^{ref} - 90\ . \tag{9}$$

Then the base station notifies the sensor node of the calculated offset "offset" and the chosen fine reference timestamp t_s^{ref}.

D. CLOCK DRIFT MONITORING

After receiving the clock offset from the base station, each sensor node infers the change in the clock offset by monitoring the transmission signal, trying to maintain constant the synchronization level achieved at t_s^{ref} (i.e., the initial synchronization error). Otherwise, the times on the sensor nodes will drift apart over time.

The duration of one transmission signal depends on the transmitted data size. In Figure 4, Equation (10) holds as mentioned in Section V-A and proven in Section VI-B.

$$\begin{cases} \Delta T_1 = \Delta T_2 = CI \\ T_{s0} - T_{b1} = T_{s2} - T_{b3} = T_{s4} - T_{b5} = T_{s6} - Tb7 = 90 \\ T_{b1} - T_{b0} = T_{b3} - T_{b2} = T_{b6} - T_{b5} \neq T_{b4} - T_{b3} \end{cases} \tag{10}$$

Therefore, $\Delta T_3 = (T_{b2} + T_{b3} - T_{b2} + 90) - (T_{b0} + T_{b1} - T_{b0} + 90) = CI$, while $\Delta T_4 = (T_{b6} + T_{b7} - T_{b6} + 90) - (T_{b4} + T_{b5} - T_{b4} + 90) \neq CI$. If we extend ΔT to cover more than one connection event, then $\frac{\Delta T}{CI} \in Z^+$ if the base station transmissions at the start and the end have the same duration. For example, $\Delta T_5 = 4 \cdot CI$ because $T_{b1} - T_{b0} = T_{b0} - T_{b8}$, and the deviation of $t_{s8} - t_{s0}$ from $4 \cdot CI$ is caused by the RTC clock drift on the sensor node.

Another issue needing consideration is the RTC timestamp increment over each CI. For CI = 100 ms, the RTC timestamp should theoretically increase by 3276.8, but instead, increases by 3276 or 3277 due to the clock resolution (clock drift not considered here). So updating the clock offset only when the RTC timestamp should increase by an integer value is suggested to avoid unnecessary fluctuation in the synchronization error.

Considering all the mechanisms mentioned above, the clock offset is updated only if the conditions in Equation (11) holds, with CI = 100 ms, N is a multiple of 5.

$$\begin{cases} \Delta T \neq \Delta t \\ \Delta T = N \cdot CI, N \in Z^+ \\ 3276.8 \cdot N \in Z^+ \end{cases} \tag{11}$$

VI. TIME SYNCHRONIZATION PERFORMANCE EVALUATION

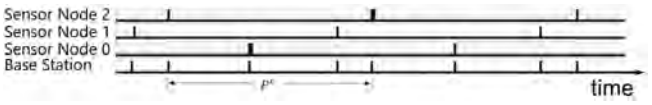
This section evaluates the described protocol for BLE time synchronization in Section V. Initially, the setup for examining the tests is described. Moreover, in the other two sections, we justify the design choices by means of measurement of the transmitted signals. Further, two sets of tests, namely toggle test and sensor test, are deployed to evaluate the time synchronization error.

A. TEST SETUP

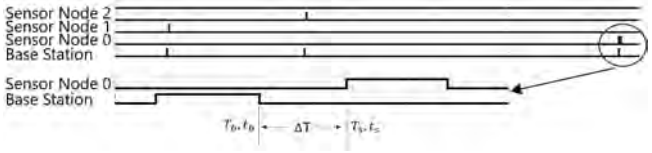
This section describes the BLE time synchronization test configuration. All the sensor nodes are synchronized to the base station RTC time after the synchronization procedure. Two tests are conducted to evaluate the accuracy of the proposed synchronization protocol, namely the toggle test and the sensor test.

In the toggle test, a meeting point time from the base station RTC reference clock is transmitted to the sensor node together with the "offset" and the picked fine reference timestamp t_s^{ref}. The sensor node converts the meeting point to the local RTC time and configures RTC channel 2 in compare mode with the converted time. A function hooked to the RTC channel 2 is called when the RTC counter register value exceeds or equals the time. In the hook function, a physical pin IOID_16 is toggled with direct register access. The output of the physical pin is sampled with a logic analyzer, and times of edge generations on different nodes are compared to calculate the synchronization error.

In the sensor test, ADC channel 1 on all the sensor nodes is fed with the same square wave signal with a frequency of 500 Hz and amplitude 1 V, generated by a DIGILENT Analog Discovery 2 to simulate the analog microphone output. When 10 pages of "microphone data" have been collected, a timestamp request flag is set. A zero-latency timer interrupt is invoked every 64 μs, i.e., the sampling period of the microphone data. In the interrupt service routine, if the timestamp request flag is set, the RTC is read, and the base station time is inferred



(a) Global time difference between two consecutive starts of a transmission signal for one sensor node at the base station's side



(b) Global time difference between the end of BLE receiver transmission and the start of sensor node transmission

FIGURE 5: A diagram showing the time differences between different transmission events from left to right over time. All the plotted signals are hooked with the tx events.

as the first element on the new page. The timestamps of the zero-crossing points of the sampled square wave signal are interpolated and compared to quantify the synchronization error.

The Texas Instrument CC1352P2 [42] is used as the BLE receiver. The transmission signal is mapped to a physical pin IOID_22 and routed to IOID_27 with a jumper wire. On each sensor node, the transmission signal is mapped to IOID_17.

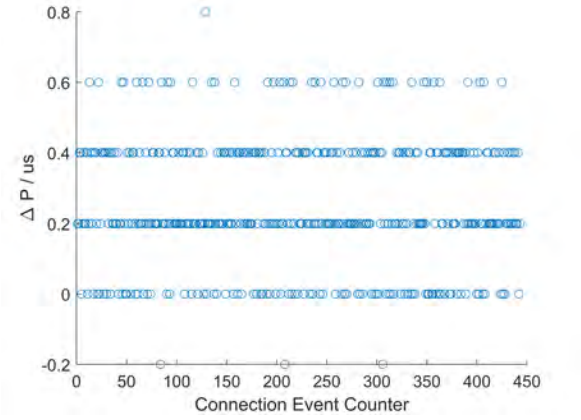
Three sensor nodes are connected to one BLE receiver with the same connection parameters, i.e., connection interval 100 ms, peripheral latency 0 , and supervision timeout 2000 ms. Nodes are distanced from one another by 0.1 m with triangular topology, while the BLE receiver is away by 0.3 m from the nodes. This distance is bounded by the measurement setup with the DIGILENT Analog Discovery 2, which needs to be physically connected with each node. Moreover, the BLE connectivity in the field has already been proven in previous studies [14], supporting up to 400 m.

B. DESIGN CHOICES AND TESTING METHODS JUSTIFICATION

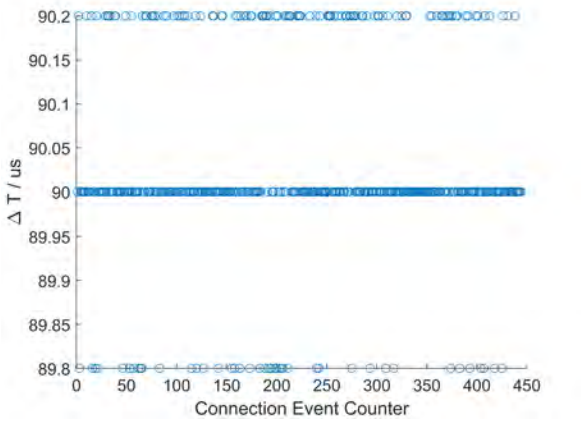
Measurements such as the transmission signals, connection events, and testing methods are conducted to justify the design choices made in Section V and the testing methods in Section VI-A. In the following, we will detail the results of these tests.

1) Transmission signal

To examine the transmission signals on the BLE receiver and the sensor node, we mapped them to physical pins and sampled them with the logic analyzer in a DIGILENT Analog Discovery 2 at 1 MHz. Let us define notations used when evaluating the transmitted signal from the node to the BLE receiver. The global time difference between the start of two consecutive transmissions for one sensor node at the base station's side is denoted with P' as shown in Figure 5a. Moreover, the difference between P' and CI is denoted with $\Delta P = P' - CI$.



(a) ΔP across connection event



(b) ΔT across connection event

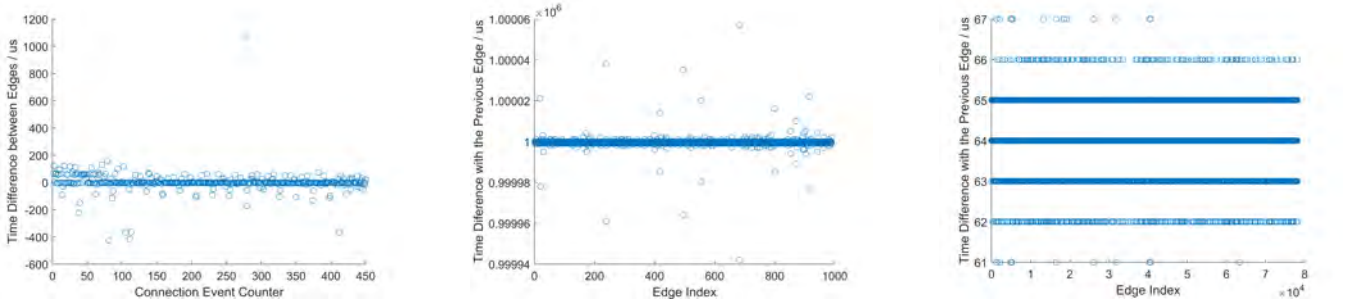
FIGURE 6: Measurement results of transmission signal for ΔP and ΔT connection event.

The global time difference between the end of the base station transmission and the start of the sensor node transmission is denoted with ΔT as shown in Figure 5b.

In Section V, we defined CI as two consecutive connection events. Thus, ideally, we should have an infinitesimal value for ΔP. Measurement results that are shown in Figure 6a indicate that ΔP reaches 0.6 μs at maximum. However, it is mostly concentrated at 0.2 μs with an overall standard deviation of 112 ns. Further, we mention that the end and the start of the sensor node transmissions are approximately spaced by 90 μs. Further, a successful coarse synchronization requires two connection events. Figure 6 shows that the measurement results for ΔT are centered at 90 μs with a standard deviation of 184 ns. This means that in a single connection event, the receiver and the transmitter have the same event.

2) Connection event

The coarse clock offset is calculated with the RTC timestamps read in connection event callbacks. To examine the calculated coarse clock offset, which is an outcome



(a) Time difference between the edges at the sensor node and base station sides.

(b) Time difference between two consecutive edges - toggle test

(c) Time difference between two consecutive edges - sensor test

FIGURE 7: Time difference between time stamps generated at the sensor node and BLE receiver at the base station for different test scenarios.

of the synchronization protocol between the receiver and the sensor node, we toggled a physical pin of the one sensor node and the BLE receiver at the base station.

Figure 7a manifests the time difference between the edges from both sides. The result shows that the generated time stamp at the sensor node and base station after time synchronization is centered at 0 after time synchronization with a standard deviation of 75.398 μs.

3) Testing methods

Two tests on the testing methods are further verified to assess inserted measurement uncertainty. For the toggle test, channel 2 is configured in continuous compare mode to generate an RTC combined interrupt every second. When the interrupt is generated, the function hooked to channel 2 is called to toggle a physical pin. The time difference between two consecutive edges and the distribution of their deviations from one second are plotted in Figure 7b. The standard deviation of the time difference is 4.126 μs. Further, the error that can possibly be introduced by the sensor test comes from the timestamping method. In particular, since the timestamps are inferred in the zero-latency interrupt service routine, the deviation of the interrupt latency affects the timestamping error. To infer this error, a physical pin is toggled with direct register access after the RTC timestamps have been read, and the time difference between two edges is demonstrated in Figure 7c. The time difference has a standard deviation of 317 ns. To conclude, the measurements verify that the proposed protocol for time synchronization is a feasible and reliable solution for real experiments with a multi-node scenario.

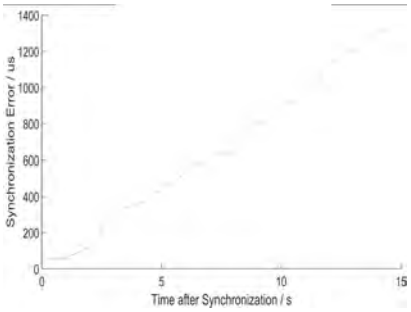
C. SYNCHRONIZATION ERROR ASSESSMENT

In the toggle test, all the sensor nodes toggle physical pins every second according to the inferred receiver time at the base station. However, the internal clock of each node suffers from drift during the sensor node acquisition time, affecting the initial synchronized system. The following results show the synchronization error in real conditions w.r.t. long-term acquisition slots, in which the internal RTC drifts significantly and/or the BLE connection event is sporadically not received Figures 8b and 8c, or in case of extreme conditions in which a synchronization update cannot happen for more than 10 s, such as Figure 8a, simulating a BLE connection failure or wireless communication absence for an extended period.

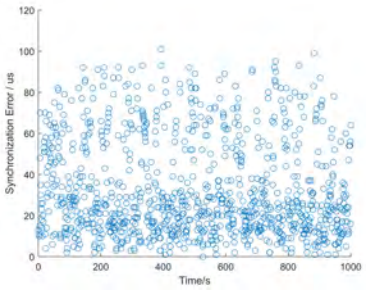
Figure 8a showcases the synchronization outcome for 15 s. Even though the synchronization error starts at 50 µs, due to clock drift between the BLE receiver at the base station and the measured node, this error monotonically increases rapidly to a range of a few ms reaching 1.4 ms at the end of the measurement. Hence, the clock drift monitoring procedure is done as described in Section V-D.

To monitor the synchronization with drift monitoring over time, toggle and sensor test measurements are repeated in a continuous mode. Notably, 10 chunks of data acquisition sessions are collected, each lasting for 100 seconds. In order to continually observe synchronization and drift monitoring, the toggling and sensor test measurements are conducted. Specifically, 10 sets of data acquisition sessions lasting for 100 seconds each are obtained.

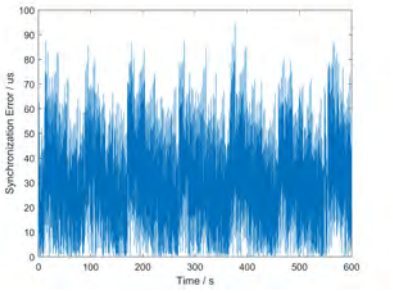
Figure 8b, Figure 8c show the result of both tests, where the effect of drift monitoring is evident in the graph, as the synchronization error remains within a few microseconds throughout the whole measurement period reaching a maximum of 100 µs, an acceptable range for the aerodynamic and acoustic measurements. Further, a snippet of the reconstructed square wave signals is presented in Figure 9 for 5 nodes connected to the BLE receiver. To sum up, the drift monitoring approach enables the synchronization error to remain steady with a maximum synchronization error 100 μs. Notably, the majority of the measured error comes from the BLE connection event uncertainty, which has been measured centered at 90 µs.



(a) Synchronization error quickly increases over time without clock drift monitoring, e.g., in case a connection event is never received after a first synchronization.



(b) Synchronization error over time in the toggle test.



(c) Synchronization error over time in the sensor test.

FIGURE 8: Synchronization error over time between time stamps generated at the sensor node and BLE receiver at the base station for different test scenarios.

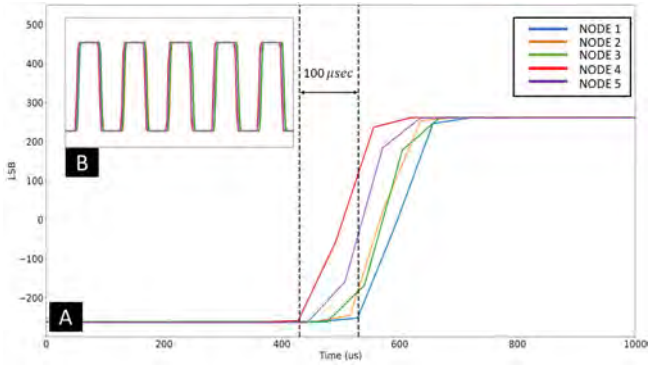


FIGURE 9: Reconstructed timestamped square wave signals at the base station.

VII. EXPERIMENTAL RESULTS ON A WIND TURBINE AND SYSTEM SCALABILITY

In this section, we examine two sets of experiments to study the scalability of the system in multi-node scenarios. First, we explore the possibility of scaling the system to deploy up to 5 nodes with a single BLE receiver, which is the maximum number of nodes supported by the limited memory of the Texas Instrument CC1352P2. However, the proposed methodology still applies to a larger number of devices within the BLE 5.1 specs. Additionally, we sweep the BLE transmission throughput to investigate the transmission time of the system in a multi-node scenario.

1) Test Setup

The microphone has a sampling rate of 16 kHz, which can be affected by the synchronization error leading to data corruption in multi-node scenarios. Two tests, namely, the scalability test and the time transmission test, are performed to examine the robustness of the system in a multi-node scenario.

The scalability tests are designed to verify the BLE bandwidth limitation in a multi-node setup. To this end, we conduct experiments using a single BLE receiver and five sensor nodes. The input to the microphones is a rectangular waveform having a period of 200 ms. Four cases are analyzed where the number of nodes increments from two to five progressively. Similar to Section VI, sensors are distanced 0.1 m apart with the same topology. For each test, an acquisition time of 30 seconds is considered to record the data from all the nodes connected to the base station.

For the time transmission test, we sweep the BLE communication bitrate from 0.5 to 2 Mbps, the transmission time of each node, hence, affecting the whole system's throughput. The system setup is the same as the scalability test. Two scenarios are considered, one involving sequential data reading and the other utilizing a parallel approach for reading the data from two or more devices. In the "Sequence Reading", we sent the "Mic-Read" command to each node one after another, starting with the first acquired node. Conversely, during "Parallel Reading" scenario testing, all sensor nodes received simultaneous commands for "Mic-Read", enabling concurrent readings from multiple nodes.

2) BLE Synchronization for Multi-node System

To extract the synchronization error between the microphone data of different nodes, we deployed the receiver at the base station, acting as a proxy to the real case scenario. Figure 9 shows the magnified reconstructed waves and the difference between them at the base station, which always fall below 100 µs. To extract the synchronization error, the rising edge of each pulse is fixed as the critical point to compare the generated time stamp for each node. Indeed, the difference between time stamps at the rising edge is considered as the synchronization error for the two nodes. Figure 10 depicts the results of the synchronization tests for varying the number of nodes connected to a single BLE receiver, reporting the average error, the variance, and the 75^{th} percentile. The synchronization error between two nodes for all the cases is between 0 to 160 µs, with a median

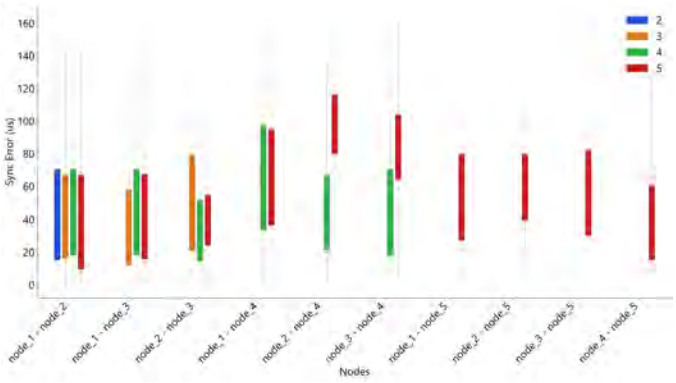


FIGURE 10: Synchronization error between a node pair in a scenario where the system is scaled from 2 to 5 nodes connected to a single base station.

error between 40 and 60 µs, providing an acceptable range for aerodynamic and acoustic systems deployed over wind turbines allowing to reconstruct signals up to a range of kHz [16], [22].

In addition, Figure 10 shows that the synchronization error concentration is at 50 µs. However, the outliers above 100 µs are due to connection BLE event drops that prevented the execution of a fine synchronization cycle, thus degrading the RTC alignment among sensor nodes that drifts over time as shown in Figure 8a. Scaling up the Aerosense system using the BLE protocol for multi-node synchronization is feasible, with an acceptable synchronization error between nodes that falls between the margins referred to above, also confirming results in Section VI-C.

3) Time Transmission and BLE Bandwidth Limitation

To examine the time transmission of the multi-node setup, we performed a BLE bandwidth variation, ranging between 0.5 to 2 Mbps. Given the test setup described in Section VII-1, each packet has a fixed size and consists of 1.952 kb (244 B), sent to the base station at each connection interval. The previously sampled sensor data, stored internally the Aerosense node on the flash memory, is therefore split in chunks of 244 B (the maximum packet size in BLE 5.1 is 250 B) and reconstructed at the receiver. Table 2 reports that in the sequence reading scenario, moving from 0.5 to 1 Mbps, the transmission time is halved as expected. However, for higher transmission bitrates, between 1 and 2 Mbps, the packet loss limits the effective transmission time of each packet, which decreases by just 4%, from 2.86 to 2.76 ms as shown in Table 2. For the same reason, if two or more nodes transmit the data to the base station simultaneously, the total BLE bandwidth at the receiver still saturates. Moreover, in this case, we also experience a high probability of connection failure, expressed with N.A. in Table 2. Notice that the only case in which the parallel packet communication was successfully carried

TABLE 2: Transmission Time per packet in the Multi-node Aerosense system in milliseconds. N.A. represent the system failures.

BLE max bitrate [Mbps]	0.5	1	1.5	2
Sequence Reading	5.13	2.86	2.85	2.76
Parallel Reading - 2 Nodes	5.70	N.A	N.A	N.A
Parallel Reading - 3 Nodes	N.A	N.A	N.A	N.A
Parallel Reading - 4 Nodes	N.A	N.A	N.A	N.A
Parallel Reading - 5 Nodes	N.A	N.A	N.A	N.A

out is with only two nodes, each of those with a transmission bitrate limited to 0.5 Mbps, suggesting that the base station can handle a data stream up to 1 Mbps, or in other words, a 1.952 kb packet every ∼2.86

For field installations, the "Sequence Reading" approach for packet transmission guarantees the system's stability. In this context, the total transmission time of the system increases linearly as the number of nodes increases. Equation (12) can be used to extract the total transmission time of the system.

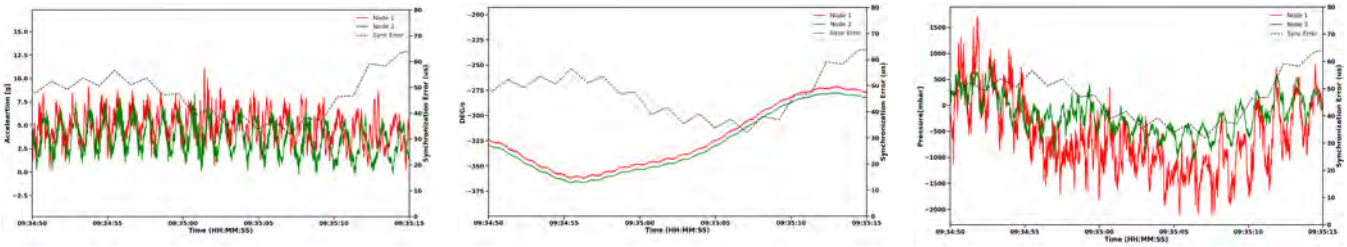
$$\text{Total}_{\text{tx}} = \text{N}_{\text{packet}} \times \text{T}_{\text{packet}} \times \text{N}_{\text{nodes}} \ . \qquad (12)$$

Where N_{packet} is the total number of packets per acquisition slot, T_{packet} is the transmission time per packet, and N_{nodes} is the total number of nodes connected to a single base station BLE receiver.

For instance, Table 1 shows that the array of microphones generates up to 3.8 Mbps while the achieved maximum throughput over the BLE wireless link is ∼1 Mbps, which can deteriorate down to 0.6 Mbps in field deployments due to packet loss and re-transmissions. Thus, the transmission time of the Aerosense system exceeds the acquisition period, with a factor that can vary between 4.2× and 7× for each connected sensor node. A transmission time with a maximum factor of 7× than acquisition time leads to a longer period between each acquisition slot. Thus, it introduces a trade-off between the period of each acquisition slot and the number of nodes connected at each base station. Finally, we can conclude that it is not advisable to connect more than five sensor nodes to the same base station but rather install a dedicated base station for each wind turbine, therefore exploiting the multi-channel feature from the BLE protocol. It not only permits multiple links to operate concurrently in the same area but also enables the use of frequency hopping, allowing concurrent data transmission from nearby wind turbines. Thereafter, the acquired measurements from sensor nodes placed on the same wind turbine or the whole wind farm can be precisely time-aligned, exploiting the precise synchronization method proposed in this paper.

VIII. FIELD AERODYNAMIC MEASUREMENTS ON AN OPERATING WIND TURBINE

The field deployment of the Aerosense system composed of two complete sensor nodes happened in June 2023 on



(a) Accelerometer raw data in [g] (b) Gyroscope raw data in [deg/s] (c) Differential pressure raw data in [mbar]

FIGURE 11: Raw data of two synchronized nodes (node 1 in red and node 2 in green) installed on the wind turbine for a wind speed of 4 m/s and rotational speed of the blade of 52 rpm.

an Aventa AV-7 [15] wind turbine. It is characterized by a rotor diameter of 12.8 m and a nominal power of 6.2 kW. As visible in Figure 12, the two sensors were installed at two different span distances on the rotor, respectively 2.5 m and 3.5 m from the tip of the blade, making them 1 m away, to monitor the aerodynamic behavior at two different spanwise positions. Further, the BLE receiver is placed at the bottom of the wind turbine's tower, approximately 18 m from the hub height. It means the height of the sensors varied between 14.5 m and 22.5 m from the ground when the blade was rotating. As visible in Figure 12, the operator installs the system on the blade surface without using any expensive crane or making any structural modifications to the wind turbine. The Aerosense sensor node is fully flexible and perfectly adheres to the blade surface, with its stability guaranteed by a waterproof adhesive. Other than demonstrating the effectiveness of the proposed measurement system in providing a cost-effective measurement solution, Figure 12 also shows a detailed view of Node 1. The photovoltaic energy harvester guarantees self-sustainability, and the main board, including the MCU and sensor arrays, is clearly visible.

Although the system can support up to 2 Mbps of data reading rate, for increasing the whole robustness of the Aerosense system in longer runs at the field, we have chosen 0.6 Mbps as the transmission bitrate.

The two sensor nodes are installed on the same blade and therefore experience the same main accelerations and rotational velocity of the wind turbine blade, as shown in Figure 11. The signals from the accelerometers (Figure 11 (a)) present both the main sinusoidal motion, which is due to the rotation of the gravity orientation in the reference frame of the blade. As both nodes have the same orientation on the same blade, the extrema in the acceleration signals happen simultaneously for both nodes, demonstrating the effective synchronization of the two nodes on an operating wind turbine, where the time synchronization error evolution over time is present on the y-left axis of Figure 11. In the 25 s window reported in Figure 11, the RTC of the two Aerosense nodes drift apart in a range between 30 µs and 65 µs.

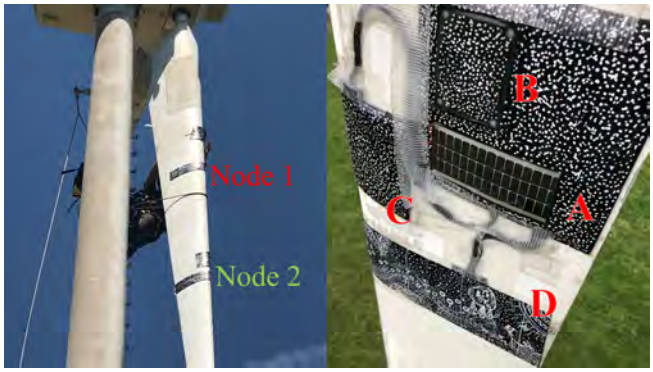


FIGURE 12: Aerosense system deployment on an operating wind turbine. A pair of identical sensors have been installed on a single blade. A zoomed view of the sensor node is provided on the right, where (A) is the photovoltaic energy harvester; (B) rechargeable battery holder; (C) five Pewatron 52 pressure sensors; (D) an array of 40 barometers ST LPS27HHW.

Moreover, the gyroscope measurements also confirm the data synchronization in the field, which measures the rotational speed of the wind turbine blade. The main change in the rotational speed is clearly superimposed for both nodes. The slight differences in the two signals are primarily due to the vibrations and deflections of the blade, which produce higher dynamics than the main rotation of the blade.

The synchronization of the two nodes makes it possible to assess the deformation of the blade by analyzing the difference between the accelerometer and gyroscope signals. In Figure 11 (c), the signals of the differential pressure sensors differ more than those from the accelerometers or gyroscopes. This is due to the turbulent atmospheric wind, whose speed and direction vary stochastically in time and space. The turbulent flow, producing the aerodynamic pressure on the blade, is not necessarily the same at both nodes. Hence, the dynamics of the pressure signals also vary between the two nodes. Severe turbulent wind conditions can cause load variations resulting in premature blade fatigue and loss of aerodynamic performance.

The synchronization of sensor nodes in the field of wind turbine monitoring is essential for fine analysis of blade motions from the accelerometers and gyroscopes,

as well as turbulent wind conditions from the high-frequency signals from the differential pressure sensors.

IX. CONCLUSION

This work presents a low-power and time-synchronized wireless sensor node for aerodynamic and acoustic measurement for large-scale multi-node wind turbine monitoring. Initially, we propose a two-step Bluetooth Low Energy synchronization protocol embedded at the MCU level. It is composed of coarse and fine synchronization, which maintains a time synchronization error within 100 µs for over 10 minutes, with a clock offset update every 500 ms in the three-node context. Further, we study the bandwidth limitation of the system up to five nodes per gateway, reaching a maximum synchronization error of 160 µs (happening concurrently with BLE packet losses) between a pair of nodes in a long-term field deployment on an operating wind turbine. Moreover, time transmission experiments in multi-node scenarios show a 600 kbps transmission bandwidth over 400 m for long-run field tests. Finally, results from two sensor nodes installed on an operative wind turbine show the robustness of the proposed protocol maintaining a synchronization error within 100 µs while sampling from a heterogeneous set of sensors on an operating wind turbine.

With the support of a multi-node installation of the Aerosense system, which is the main contribution of this paper, future enhancements in wind energy generation will be enabled. Indeed, collecting aerodynamic samples directly from an operating wind turbine enables a more detailed understanding of three-dimensional turbulent flow over rotor blades, supporting several case studies with real-time requirements. For instance, using local surface measurements to infer the blade AoA and rotor inflow conditions helps manufacturers to improve design and simulation tools. Moreover, the acquired measurements and operational history can also increase revenues by improving decision-making and asset management of sub-optimal control settings. For instance, the acoustic data can be exploited to detect and subsequently decrease the audio noise emissions, which is known to upset the residents and, thus, increase wind energy acceptance. Furthermore, the Aerosense system can enable early detection and classification of local damage or deterioration, reducing operating costs by improving operators' decision-making regarding blade maintenance and repair.

ACKNOWLEDGMENT

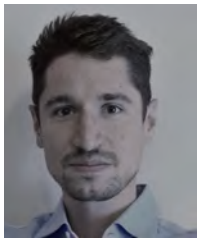
The authors sincerely thanks the Aerosense[2] group and several people that supported the realisation of this paper. In particular, Sarah Barber, Yurij Marykovskiy, Manuel Strahm, Stefano Ballarin Dolfin, and Tom Clark.

[2]www.aerosense.ai

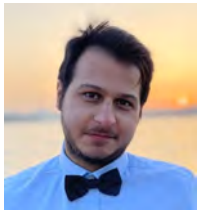
REFERENCES

[1] D. I. Armstrong McKay, A. Staal, J. F. Abrams et al., "Exceeding 1.5 c global warming could trigger multiple climate tipping points," Science, vol. 377, no. 6611, p. eabn7950, 2022.
[2] H. L. Van Soest, M. G. den Elzen, and D. P. van Vuuren, "Net-zero emission targets for major emitting countries consistent with the paris agreement," Nature communications, vol. 12, no. 1, p. 2140, 2021.
[3] D. Y. Leung and Y. Yang, "Wind energy development and its environmental impact: A review," Renewable and sustainable energy reviews, vol. 16, no. 1, pp. 1031–1039, 2012.
[4] S. D. Ahmed, F. S. Al-Ismail, M. Shafiullah et al., "Grid integration challenges of wind energy: A review," IEEE Access, vol. 8, pp. 10 857–10 878, 2020.
[5] M. Fahim, V. Sharma, T.-V. Cao et al., "Machine learning-based digital twin for predictive modeling in wind turbines," IEEE Access, vol. 10, pp. 14 184–14 194, 2022.
[6] Y. Teff-Seker, O. Berger-Tal, Y. Lehnardt et al., "Noise pollution from wind turbines and its effects on wildlife: A cross-national analysis of current policies and planning regulations," Renewable and Sustainable Energy Reviews, vol. 168, p. 112801, 2022.
[7] N. Huang, Q. Chen, G. Cai et al., "Fault diagnosis of bearing in wind turbine gearbox under actual operating conditions driven by limited data with noise labels," IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1–10, 2020.
[8] R. Fischer, H. Mueller, T. Polonelli et al., "Windnode: A long-lasting and long-range bluetooth wireless sensor node for pressure and acoustic monitoring on wind turbines," in 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS). IEEE, 2021, pp. 393–399.
[9] F. Di Nuzzo, D. Brunelli, T. Polonelli et al., "Structural health monitoring system with narrowband iot and mems sensors," IEEE Sensors Journal, vol. 21, no. 14, pp. 16 371–16 380, 2021.
[10] X. Jin, Z. Xu, and W. Qiao, "Condition monitoring of wind turbine generators using scada data analysis," IEEE Transactions on Sustainable Energy, vol. 12, no. 1, pp. 202–210, 2020.
[11] M. Mieloszyk and W. Ostachowicz, "An application of structural health monitoring system based on fbg sensors to offshore wind turbine support structure model," Marine Structures, vol. 51, pp. 65–86, 2017.
[12] H. Aagaard Madsen, C. Bak, U. Schmidt Paulsen et al., The DAN-AERO MW experiments. Final report, ser. Denmark. Forskningscenter Risoe. Risoe-R. Danmarks Tekniske Universitet, Risø Nationallaboratoriet for Bæredygtig Energi, 2010, no. 1726(EN).
[13] S. Barber, J. Deparday, Y. Marykovskiy et al., "Development of a wireless, non-intrusive, mems-based pressure and acoustic measurement system for large-scale operating wind turbine blades," Wind Energy Science, vol. 7, no. 4, pp. 1383–1398, 2022.
[14] T. Polonelli, H. Müller, W. Kong et al., "Aerosense: A self-sustainable and long-range bluetooth wireless sensor node for aerodynamic and aeroacoustic monitoring on wind turbines," IEEE Sensors Journal, vol. 23, no. 1, pp. 715–723, 2022.
[15] A. AG. ([Online] 2023) Aventa av-7 technische daten. (accessed: 01.10.2023). [Online]. Available: www.wind-turbine-models.com/turbines/1529-aventa-av-7
[16] J. Deparday, H. Müller, T. Polonelli et al., "An experimental system to acquire aeroacoustic properties on wind turbine blades," in Journal of Physics: Conference Series, vol. 2265, no. 2. IOP Publishing, 2022, p. 022089.
[17] R. Soto-Valle, S. Bartholomay, J. Alber et al., "Determination of the angle of attack on a research wind turbine rotor blade using surface pressure measurements," Wind Energy Science, vol. 5, no. 4, pp. 1771–1792, 2020.
[18] K. M. Fathima, R. S. Raj, K. R. Prasad et al., "Mems multi sensor intelligent damage detection for wind turbines by using iot," in Journal of Physics: Conference Series, vol. 1916, no. 012045. IOP Publishing, 2021.

[19] C. Raab and K. Rohde-Brandenburger, "In-flight testing of mems pressure sensors for flight loads determination," in AIAA Scitech 2020 Forum, no. 0512, 2020.
[20] J. Filipskỳ, J. Čížek, F. Wittmeier et al., "Design and first test of the new synchronous 200 hz system for unsteady pressure field measurement," in FKFS Conference. Springer, 2017, pp. 252–263.
[21] S. Karad and R. Thakur, "Efficient monitoring and control of wind energy conversion systems using internet of things (iot): a comprehensive review," Environment, Development and Sustainability, vol. 23, 10 2021.
[22] T. Polonelli, J. Deparday, I. Abdallah et al., "Instrumentation and measurement systems: Aerosense: A wireless, non-intrusive, flexible, and mems-based aerodynamic and acoustic measurement system for operating wind turbines," IEEE Instrumentation & Measurement Magazine, vol. 26, no. 4, pp. 12–18, 2023.
[23] B. Wondra, S. Malek, M. Botz et al., "Wireless high-resolution acceleration measurements for structural health monitoring of wind turbine towers," Data-Enabled Discovery and Applications, vol. 3, no. 1, p. 4, 2019.
[24] L. Lu, Y. He, T. Wang et al., "Wind turbine planetary gearbox fault diagnosis based on self-powered wireless sensor and deep learning approach," IEEE Access, vol. 7, pp. 119 430–119 442, 2019.
[25] F. Berkemeyer and W. Lang, "Wireless inertial measurements on wind turbine rotor blades," IEEE Sensors Journal, vol. 21, no. 24, pp. 27 938–27 946, 2021.
[26] L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," IEEE Internet of Things Journal, vol. 7, no. 1, pp. 16–32, 2019.
[27] D. Mitchell, J. Blanche, O. Zaki et al., "Symbiotic system of systems design for safe and resilient autonomous robotics in offshore wind farms," IEEE Access, vol. 9, pp. 141 421–141 452, 2021.
[28] I. Zhou, I. Makhdoom, N. Shariati et al., "Internet of things 2.0: Concepts, applications, and future directions," IEEE Access, vol. 9, pp. 70 961–71 012, 2021.
[29] S. N. Swamy and S. R. Kota, "An empirical study on system level aspects of internet of things (iot)," IEEE Access, vol. 8, pp. 188 082–188 134, 2020.
[30] Z. Idrees, J. Granados, Y. Sun et al., "Ieee 1588 for clock synchronization in industrial iot and related applications: a review on contributing technologies, protocols and enhancement methodologies," IEEE Access, vol. 8, pp. 155 660–155 678, 2020.
[31] F. Asgarian and K. Najafi, "Bluesync: Time synchronization in bluetooth low energy with energy-efficient calculations," IEEE Internet of Things Journal, vol. 9, no. 11, pp. 8633–8645, 2021.
[32] S. Sridhar, P. Misra, G. S. Gill et al., "Cheepsync: a time synchronization service for resource constrained bluetooth le advertisers," IEEE Communications Magazine, vol. 54, no. 1, pp. 136–143, 2016.
[33] F. Asgarian and K. Najafi, "Bluesync: Time synchronization in bluetooth low energy with energy efficient calculations," IEEE Internet of Things Journal, 2021.
[34] A. Comuniello, A. De Angelis, A. Moschitta et al., "Using bluetooth low energy technology to perform tof-based positioning," Electronics, vol. 11, no. 1, p. 111, 2021.
[35] C. C. Rheinländer and N. Wehn, "Precise synchronization time stamp generation for bluetooth low energy," in 2016 IEEE SENSORS. IEEE, 2016, pp. 1–3.
[36] A. Yousefi, K. Somaratne, and F. J. Dian, "Analysis of time synchronization based on current measurement for bluetooth low energy (ble)," in 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON). IEEE, 2017, pp. 602–607.
[37] F. J. Dian, A. Yousefi, and K. Somaratne, "A study in accuracy of time synchronization of ble devices using connection-based event," in 2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2017, pp. 595–601.
[38] G. Biagetti, P. Crippa, L. Falaschetti et al., "A multi-channel electromyography, electrocardiography and inertial wireless sensor module using bluetooth low-energy," Electronics, vol. 9, no. 6, p. 934, 2020.
[39] M. Harada, S. Izumi, R. Kozeni et al., "20-μs accuracy time-synchronization method using bluetooth low energy for internet-of-things sensors," in 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2022, pp. 181–186.
[40] C. Bak, F. Zahle, R. Bitsche et al., "The dtu 10-mw reference wind turbine," in Danish Wind Power Research 2013, 2013.
[41] H. Madsen, P. Fuglsang, J. Romblad et al., "The dan-aero mw experiments," in 48th Aiaa Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2010, p. 645.
[42] Texas Instruments, CC1352P Data Sheet, Texas Instruments Incorporated, Dallas, Texas 75265 USA, 2021, accessed October 01 2023. [Online]. Available: www.ti.com/lit/ds/symlink/cc1352p.pdf?ts=1649748851348&ref_url=https



TOMMASO POLONELLI (M'20) received the M.Sc. degree and the Ph.D. in electronics engineering from the University of Bologna, Bologna, Italy, in 2017 and 2020, respectively. He is currently a Lecturer and post-doctoral researcher at ETH Zürich, Zürich, Switzerland. His research work focuses on wireless sensor networks, IoT, autonomous unmanned vehicles, power management techniques, structural health monitoring, and the design of ultra-low power battery-supplied devices with onboard intelligence. He has collaborated with several universities and research centers, such as the University College Cork, Cork, Ireland, and the Imperial College London, London, U.K. He has authored over 30 papers in international journals and conferences.



AMIRHOSSEIN MOALLEMI received the B.Sc. degree in electrical engineering, electronics from Zanjan University, Zanjan, Iran, in 2017, and the M.Sc. degree in electronic engineering from the University of Bologna, Bologna, Italy, in 2020, where he is currently pursuing the PhD degree with the Department of Electrical, Electronic and Information Technologies Engineering (DEI). His research interests include IoT, low-power hardware and firmware design for embedded systems, machine and deep learning models, and structural health monitoring systems.

HANNA MÜLLER Hanna Müller received the B.Sc. and M.Sc. degrees in electrical engineering and information technologies from ETH Zürich, Zürich, Switzerland, in 2017 and 2020, respectively, where she is currently pursuing the Ph.D. degree with the Integrated Systems Laboratory. Her research interests include low-power systems, wireless sensor networks and onboard intelligence.

WEIKANG KONG received his M.Sc. degree in Electrical Engineering and Information Technology from ETH Zürich, Zürich, Switzerland in 2022. His research interests include energy harvesting and wireless embedded systems.

JULIEN DEPARDAY received his M.Sc degree in mechanics from the ENSTA-Bretagne (Brest, France) in 2010 and his Ph.D degree from the Université Bretagne Occidentale (Brest, France) in 2016 in fluid mechanics. He is currently a senior scientist at OST, Eastern Switzerland University of Applied Sciences, at the Institute for Energy Technology. He is specialized in experimental fluid dynamics, with a strong emphasis on field-measurements and unsteady aerodynamics. He developed several in-field measurement systems, implying interdisciplinary projects with electronics experts. His research interests are in fluid-structure interactions, flow separation, and inviscid flow theory for naval engineering, wind energy and aeronautics.

LUCA BENINI holds the chair of digital Circuits and systems at ETHZ and is Full Professor at the Università di Bologna. He received a PhD from Stanford University. Dr. Benini's research interests are in energy-efficient parallel computing systems, smart sensing micro-systems and machine learning hardware. He is a Fellow of the ACM and a member of the Academia Europaea. He is the recipient of the 2016 IEEE CAS Mac Van Valkenburg award, the 2020 EDAA achievement Award and the 2020 ACM/IEEE A. Richard Newton Award.

MICHELE MAGNO is currently a Senior Scientist at ETH Zürich, Switzerland, at the Department of Information Technology and Electrical Engineering (D-ITET). Since 2020, he is leading the D-ITET center for project-based learning at ETH. He received his master's and Ph.D. degrees in electronic engineering from the University of Bologna, Italy, in 2004 and 2010, respectively. He is working in ETH since 2013 and has become a visiting lecturer or professor at several universities, namely the University of Nice Sophia, France, Enssat Lannion, France, Univerisity of Bologna and Mid University Sweden, where currently is a full visiting professor at the electrical engineering department. His current research interests include smart sensing, low-power machine learning, wireless sensor networks, wearable devices, energy harvesting, low-power management techniques, and extension of the lifetime of batteries-operating devices. He has authored more than 220 papers in international journals and conferences. He is a senior IEEE member and an ACM member. Some of his publications were awarded as best papers awards at IEEE conferences. He also received awards for industrial projects or patents.