IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 16, 2021, accepted March 4, 2021, date of publication March 12, 2021, date of current version March 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3065775

Singular Spectrum Analysis for Source Separation in Drone-Based Audio Recording

FRANCISCO GARCÍA ENCINAS[1], LUÍS AUGUSTO SILVA[1], ANDRÉ SALES MENDES[1], GABRIEL VILLARRUBIA GONZÁLEZ[1], VALDERI REIS QUIETINHO LEITHARDT[2,3], (Member, IEEE), AND JUAN FRANCISCO DE PAZ SANTANA[1]

[1]Expert Systems and Applications Laboratory (ESALAB), Faculty of Science, University of Salamanca, 37008 Salamanca, Spain
[2]COPELABS, Universidade Lusófona de Humanidades e Tecnologias, Campo Grande 376, 1749-024 Lisboa, Portugal
[3]VALORIZA, Research Center for Endogenous Resources Valorization, Instituto Politécnico de Portalegre, 7300-555 Portalegre, Portugal

Corresponding authors: Francisco García Encinas (frangaren@usal.es) and Luís Augusto Silva (luisaugustos@usal.es)

This work was supported by the Junta Castilla y León through the Project Actividades en el Proyecto Desarrollo de Tecnología de Registro de Niveles de Ruido Mediante el Empleo de Drones under Grant 2020/00006/001-L704. The work of Francisco García Encinas was supported by the Spanish Ministry of Education and Vocational Training through the FPU Fellowship under Grant FPU19/02455. The work of André Sales Mendes was supported in part by the European Social Fund, and in part by the Junta de Castilla y León (Operational Programme 2014–2020 for Castilla y León, EDU/556/2019 BOCYL).

ABSTRACT The usage of drones is increasingly spreading into new fields of application, ranging from agriculture to security. One of these new applications is sound recording in areas of difficult access. The challenge that arises when using drones for this purpose is that the sound of the recorded sources must be separated from the noise produced by the drone. The intensity of the noise emitted by the drone depends on several factors such as engine power, propeller rotation speed, or propeller type. Noise reduction is thus one of the greatest challenges for the next generations of unmanned aerial vehicles (UAVs) and unmanned aerial systems (UAS). Even though some advances have been made on that matter, drones still produce a considerable noise. In this article, we approach the problem of removing drone noise from single-channel audio recordings using blind source separation (BSS) techniques, and in particular, the singular spectrum analysis algorithm (SSA). Furthermore, we propose an optimization of this algorithm with a spatial complexity of $\mathcal{O}(nt)$, which is significantly lower than the naive implementation which has a spatial complexity of $O(tk^2)$ (where n is the number of sounds to be recovered, t is the signal length and k is the window size). The best value for each parameter (window length and number of components used to reconstruct the source) is selected by testing a wide range of values on different noise-sound ratios. Our system can greatly reduce the noise produced by the drone on said recordings. On average, after the recording has been processed by our method, the noise is reduced by 1.41 decibels.

INDEX TERMS Drone, audio recording, source separation, egonoise cancellation, singular spectrum analysis.

I. INTRODUCTION

Unmanned Aerial Vehicles, also known as drones, are an emerging technology used by both civilians and the military. In fact, the popularity of this kind of vehicle has increased so much, that it is no longer used exclusively for work, but also for leisure by civilians. This democratization has been enabled by the production of cheaper, more accurate, and easier to use drones [1]. Drones have also helped on tasks that a few years ago were very complex to perform, e.g, controlling forest fires, locating missing people after natural disasters, agriculture [2], or even road monitoring [3]. However, even though they are useful for many tasks, remote control planes have certain problems that can restrict their operation. One of the most significant is the noise produced by the propellers. This disadvantage can render the drone unsuitable in situations where noise regulations exist or when high noise emissions can adversely affect the task carried out. For example, many airports around the world impose strict limitations on the level of noise allowed at daytime or

The associate editor coordinating the review of this manuscript and approving it for publication was Xiwang Dong.

nighttime, and the acoustic signature of a military aircraft has a significant impact on whether it can go unnoticed. The importance of the acoustic signature of propeller-powered air vehicles was noted as early as the 1960s [4] and led to the development of the YO-3A Quiet Star, which later became an acoustic research aircraft [5]. Today, due to the increased usage of propeller-based unmanned aerial vehicles, there is a renewed interest in reducing the noise produced by the propeller [6]; however, the vehicles are still far from silent. Some industries, e.g., noise control or cinematic and audiovisual, would require to remove most, if not all, of the noise in captured in the recordings. A potential strategy to reaching this goal would be to use software algorithms that reduce noise on already recorded sounds. The present paper is centered around this very idea. More specifically, we use singular spectrum analysis to reduce the noise to a great extent.

Industrial noise in the vicinity of residential buildings is limited by certain thresholds defined in regulations. If the noise level emitted by an industrial complex is above that particular limit, the main noise sources must be relocated in order to ensure noise reduction [7]. The identification of such sources within an industrial complex is a key issue in the noise reduction process and can be done with the help of modern tools such as acoustic cameras. Acoustic photos or videos show which are the dominant noise sources at each observation point. It is well known that sources which produce a higher level of noise have a greater impact on the overall radiation of the noise in a defined direction [8].

Companies specialized in noise control seek to make measurements in several places, such as factories, chimneys, and machines that generate much noise and vibration. To measure the noise, human operators have to approach the sources. This process is risky, because it usually involves raising the operator to great heights and staying close to dangerous industrial machines. For example, measuring the noise of an industrial plant with buildings higher than 50 meters could involve climbing the ladders in its chimneys and using lifting platforms to get close to certain machines. With the system proposed in this work, human workers would not need to risk their lives by trying to reach dangerous inaccessible spots on these industrial complexes.

In this article, we study how to reduce drone-noise from single-channel recordings. This is seen as a blind source separation (BSS) problem [9]. We thus approach it using the singular spectrum analysis (SSA) algorithm [10] which, in contrast to other methods such as active noise canceling [11], only needs one observation. Furthermore, we propose an optimization for this algorithm with a spatial complexity of $\mathcal{O}(nt)$, which is much lower than the naive implementation with a spatial complexity of $O(tk^2)$ (where n is the number of sounds to be recovered, t is the signal length and k is the window size). This optimization enables us to process longer time series and test a greater number of configurations. Lastly, we propose an architecture for a drone-based recording system. We have validated the results of the noise reduction by applying different configurations of SSA over different synthetic observations generated from recordings in an anechoic chamber. The results were verified by measuring the time elapsed and memory consumed for different parameters of the algorithm on both the optimized version and the unoptimized one.

The rest of the paper is organized as follows. Section 2 introduces the background and related works. Section 3 introduces the drone application, the proposed architecture, and the optimization of the algorithm. Sections 4 and 5 present the experimental results, discuss the blind source separation, and provide a conclusion, respectively.

II. BACKGROUND

In this section, we analyze different works carried out in the areas involved. To provide a more detailed and clear analysis, the background of each of the research areas is expounded in a different subsection. In the first of them (section II-A), we begin by explaining the ego-noise cancellation problem and passive noise cancellation methods used to mitigate it. Then, we focus on software algorithms and digital filters able to reduce noise in signals (section II-B). Finally, blind source separation-based methods are reviewed more thoroughly, as this is the approach used in our framework (section II-C). All the methods explained in this section are summarized in Table 1.

A. DRONE NOISE AND EGO-NOISE REDUCTION

The question this research seeks to answer is how it is possible to record propeller noise-free audio using unmanned aerial vehicles. This self-produced noise cancellation problem, which the literaure presents as ego noise reduction, is not restricted to this domain of application. Cars and robots equipped with microphones are also exposed to several sources of noise, e.g., engines, motors, propellers, and rotating joints, among other mechanical components and moving parts. The sounds produced by these severely corrupt recorded signals and harm the robot's ability to react to unforeseen acoustic events. Given the ubiquity of noise sources, ego noise reduction is a crucial problem in the path towards robot hearing [39], [40].

Ego noise reduction is a particularly difficult problem for several reasons. First, given that ego-noise sources are usually closer to the microphones, self-produced noise can be louder than other signals of interest, e.g., speech. Suppose a drone with engines that produce a 65db sound on its current regime of operation were measuring the noise level of a large chimney. Even if the chimney produces a sound with the same power, that sound would appear quieter, because the engines are only a few inches away from the recording devices. Another challenging aspect is the difficulty to model the ego noise. For example, drones have different engine configurations, using 4, 6, or 8 of them. In turn, each of these configurations has different acoustic characteristics. Moreover, the ego-noise is highly non-stationary, as it typically depends

TABLE 1. Noise reduction algorithms summary.

Noise Reduction	Passive (materials and drone design)				[12]–[14]
	Algorithmic	Adaptive	Wiener filter		[15]–[17]
			Kalman filter		[18], [19]
			Neural network		[20]–[22]
			Evolutionary algorithm		[23], [24]
		Non-adaptive	Low-pass/high-pass/pass-band filter		[25]
			Wavelet transform		[26]–[28]
			Beam-forming		[29]
			Singular value decomposition	Independent component analysis	[30]
				Principal component analysis	[31], [32]
				Singular spectrum analysis	[33]–[35]
				Other	[36]–[38]

on the characteristics of the movements being performed, e.g., speeds and accelerations.

The noise produced by a drone has three main components, namely, the mechanical noise generated by the rotation of the motors, the noise generated by the propellers cutting through the air, and the noise of the airflow generated by the propellers themselves. The airflow generated by the propellers has a downward direction, and microphone placement has a great impact on how the recording system is affected. Typically, the noise is heavily mitigated when placing the microphones on or beside the drone instead of below the propellers [12]. Furthermore, propeller noise can be reduced by using a more optimal shape [13], [14] or a noise-conscious control system [41]. However, even though the noise generated by the propellers and their airflow can be heavily mitigated with clever microphone placement and better propeller shapes, there is still some remnant noise that needs to be removed from the recording using software-based alternatives.

B. SOFTWARE-BASED NOISE REDUCTION

The techniques portrayed in subsection II-A had to be applied before or while recording, for example by replacing the propellers with more appropriate ones or using specific control strategies during the flight. Although the noise emitted could be heavily reduced using these kinds of techniques, some of it was still captured on the recordings. There are several filters described in the literature that can process these recordings and approximate the noiseless signal.

Filtering algorithms can be divided into two different categories: adaptive methods [42] and non-adaptive methods. On the one hand, adaptive methods use optimization algorithms to adjust their parameters to the signal on hand. These methods are capable of learning characteristics about the signal and the noise and improve their performance when training. The Wiener filter, neural networks, and evolutionary algorithms [23], [24] are examples of adaptive filters. On the other hand, non-adaptive methods do not use optimization algorithms. In the same way as adaptive methods, they also have parameters. However, their value is chosen based on empirical experimentation and prior knowledge, in contrast with the optimization process carried on adaptive methods. Examples of non-adaptive methods are wavelet transform based [26]–[28], singular value decomposition based [36]–[38], or, even pass-band filters.

Neural network-based approaches can achieve state-of-the-art results in application domains with great amounts of data. RNNoise [20], put forward by Zhang *et al.* [21], and the Wang *et al.* method [22] are both great alternatives in the voice denoising field. However, they are not as performant in domains with much less data. Collecting such amounts of data in a specific application domain is, at best, a very costly and time-consuming process. It can even become impossible in cases where the nature of the data is unknown beforehand.

Classical adaptive filters do not need as much data as neural networks. Several works use Kalman [18], [19] or Wiener filters [15]–[17] successfully to denoise different types of signals. However, such filters have their limitations as well. On the one hand, Wiener filters perform great when dealing with stationary Gaussian noise. On the other hand, when facing a non-stationary noise, Kalman filters and the like are superior. However, Kalman filters assume Gaussian distributed noise and are not very robust against modeling errors.

In some cases, when the source or the noise signals are located in a very specific zone of the frequency spectrum, the problem can be easily solved using simple filters such as high-pass, low-pass [25], pass-band, or band-stop. Nevertheless, due to the broadband and highly non-stationary nature of the ego-noise, these measures are not applicable.

Singular spectrum analysis, the method used in this work, is a singular value decomposition-based method. This method can decompose a signal into a chosen number of components. These components can then be associated with different sources or data trends. And, since this extraction is done without any prior knowledge concerning the source signals, singular spectrum analysis is said to be a blind source separation method.

C. SINGULAR SPECTRUM ANALYSIS AND BLIND SOURCE SEPARATION METHODS

During the past decades, much attention has been paid to the separation of mixed sources, in particular to the blind cases where both the sources and the mixing process are unknown, and only the recordings of the mixtures are available. In several situations, it is desirable to recover all sources from the recorded mixtures, or at least to segregate a particular source. Furthermore, it may be useful to identify the mixing process itself in order to uncover information about the physical mixing system [10], [43].

Wang and Cavallaro [9] proposes a BSS frame that extracts a target sound from the noisy multi-channel signals picked up by a set of microphones mounted on a drone. Thus, the frame improves the sound by treating the target and noise signals equally and separating the sources from the mixed signals captured by the microphone array [44].

Blind source separation consists in recovering the signals $\mathbf{S}$ emitted by n sources from observations $\mathbf{X}$ obtained at t time instants by m sensors that introduce an additive noise $\mathbf{N}$. Assuming that the observations have been generated from the signals utilizing a linear mixing system $\mathbf{A}$, the problem can be mathematically formalized in (1). Note that $\mathbf{S}$ is a $n \times t$ matrix, $\mathbf{X}$ is a $m \times t$ matrix, $\mathbf{A}$ is a $m \times n$ matrix, $\mathbf{N}$ is a $m \times t$ matrix, and m, n, and t are scalars.

$$\mathbf{X} = \mathbf{AS} + \mathbf{N} \tag{1}$$

The solution to this problem consists of finding a $n \times m$ matrix $\mathbf{W}$, named unmixing matrix, such that it allows to approximate the signals emitted by the sources from the observations ($\hat{\mathbf{S}}$ matrix in this approach), as shown in (2).

$$\hat{\mathbf{S}} = \mathbf{WX} \approx \mathbf{S} \tag{2}$$

Finding $\mathbf{W}$ when the mixing system $\mathbf{A}$ is known is trivial. For example, when $n = m$, $\mathbf{W} \approx \mathbf{A}^{-1}$. However, the mixing system is usually unknown. For this reason, it is necessary to use certain mechanisms to find the unmixing matrix $\mathbf{W}$ based solely on the observations $\mathbf{X}$.

Most jobs use more sensors than the number of sources. Techniques like beamforming [29], independent component analysis [30] or principal component analysis [31], [32] are also used to recover the signals emitted by the sources from the observations. However, when the number of observations is smaller than the number of sources ($m < n$), the problem is ill-posed and thus the difficulty is much greater. Some methods obtain results using singular spectrum analysis or adaptive filters. In many cases, making assumptions about the sources can help to improve the results. For example, when the sources are assumed to be human voices, there is a greater variety of methods at one's disposal. However, it is not always possible to limit the type of sources in this way.

In this work, singular spectrum analysis is used to denoise the sound recorded by the drone. This choice is justified by some advantages that this method presents in comparison with others. One of them is that singular spectrum analysis does not require prior knowledge concerning the source signals. This is very important, because the system should be able to reduce noise for many different types of sources. Additionally, the system can achieve great results using a single observation from a single microphone.

III. PROPOSED METHOD

The proposed system can separate the sound produced by a drone from the sound of an unknown source on a mixture recorded with a single microphone. This task is approached as a blind source separation problem, with $m = 1$ and $n = 2$. As such, the singular spectrum analysis [43] algorithm is used to extract the principal components of a signal and reconstruct the drone and the source sounds based on these components. The method is explained in more detail in subsection III-B.

The recordings of the drone are uploaded to a cloud database using 4G technology. They are later processed in a cloud server using singular spectrum analysis to retrieve the principal components of a signal and later mix them to reconstruct the original sounds. The drone is capable of working in a semi-autonomous way due to a way-point system and, if needed, it can also be teleoperated in real-time by an operator. The proposed architecture is described in subsection III-A.

Our system can carry out this process with a very low memory footprint due to the optimization proposed in subsection III-C of this article. In summary, our method exploits the redundancy of the elements of Hankel matrices to substitute some matrix product operations for a convolution of vectors. This change greatly decreases the memory complexity of the algorithm from $\mathcal{O}\left(tk^2\right)$ to $\mathcal{O}(nt)$ where t is the signal length, k the window length, and n the number of sources to reconstruct (note that $n \leq k$). The time complexity of the optimized algorithm is also lower. These improvements are validated empirically in the results section.

A. PROPOSED ARCHITECTURE

In this section, the proposed architecture will be presented as a solution to the problem at hand. The architecture aims to address the various issues that these systems have in terms of communication, route planning, synchronization, and signal processing. The main feature of this architecture is the ability to cope with and include new features, as well as adapt to new environments in a simple way. The architecture proposed for this research article is presented in Fig. 1.

The proposed architecture is divided into several modules:

- *Cloud Services:* this module refers to the services deployed in the cloud, in which the information and data storage subsystems are located. The system generates different types of data: report data, time series, audio, and video files. Different types of databases have been used to face this challenge. For the storage of video and audio files, the server's own file system is used, keeping references such as synchronization time stamps in databases. For the storage of data that can be classified as time series InfluxDB is used, which is a database optimized for this type of data so that you can operate and group data records efficiently. In this database, you can find the drone telemetry data such as altitude, GPS coordinates, orientation, etc. For other application data that are not considered as time series, a MongoDB non-relational database is used. This module also displays the drone noise reduction post-processing services for environmental analysis. The data exchange occurs by means of an API which has been implemented to allow communication from both drone and reports analysis applications. Real-time telemetry data is exchanged between the drone and the system thanks to the MQTT

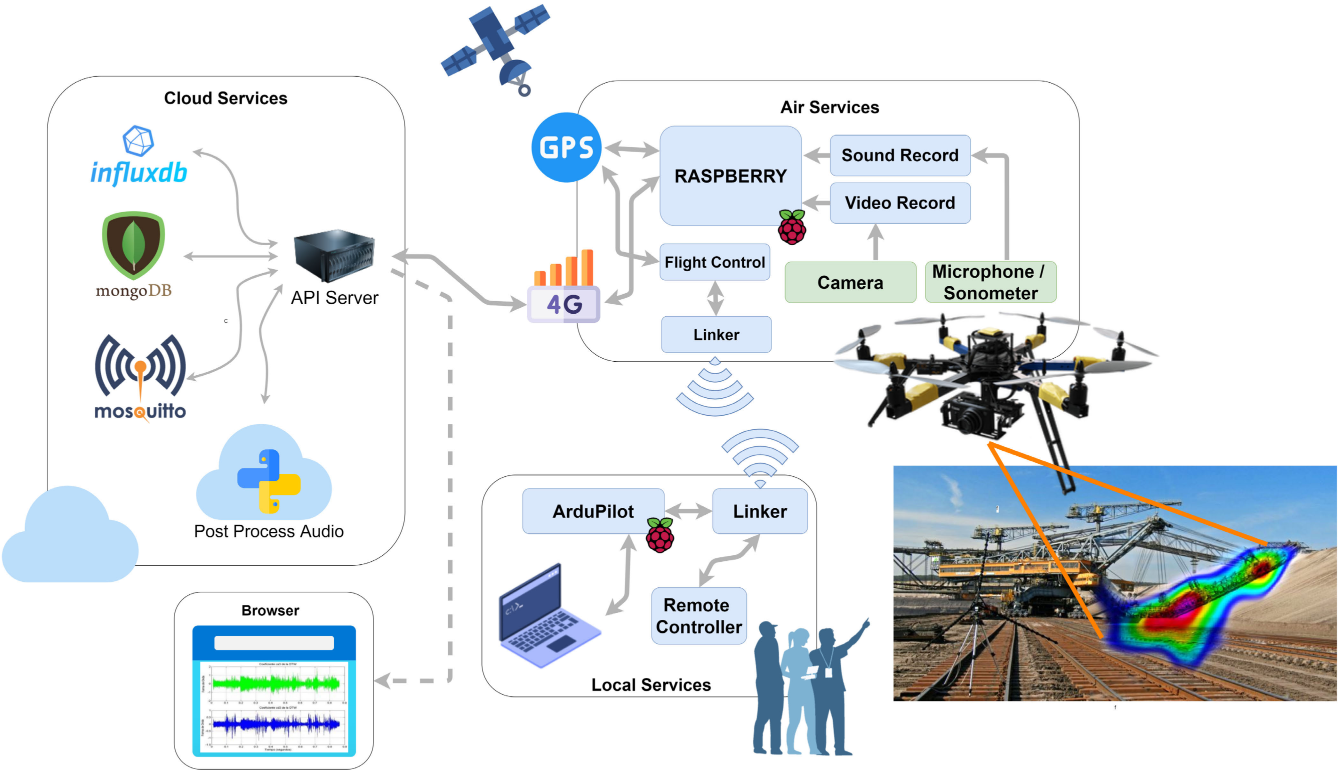

FIGURE 1. Proposed Architecture.

protocol, which is used with a mosquito server that allows the exchange of information with minimum delay and data consumption.

- *Air Services:* this module refers to the services and devices deployed in the drone which are necessary to inspect the environment, data collection, and communication. The transmission of data to the server takes place via 4G technology, as this is the easiest way to connect to the Internet when the drone is at a relatively high distance from the ground. The 4G connection can be used in this case study because it focuses on monitoring industrial environments, and we usually have coverage in these environments. For the sensorization of the environment, the drone is equipped with a video camera that allows the analysis of noise sources. To capture the ambient noise, we propose a directional microphone or a sound level meter that will ensure a reliable audio source with high sampling characteristics. The video and audio sources are stored in the local Raspberry Pi and transmitted to the server for post-processing.
- *Local Services:* this module refers to route planning and flight supervision. We propose using the Ardupilot for route planning. This open-source software allows route planning by indicating waypoints and the height we want the drone to reach during its flight. This system can integrate multiple devices and use notification services such as [45]. In this module, the user can supervise the flight by watching the live video being captured by the drone and taking remote control in case of an emergency.

Finally, the architecture includes a web interface that allows us to see the results of the audio analysis and the synchronized videos in order for the noise sources to be detected.

B. SINGULAR SPECTRUM ANALYSIS

Let $\mathbf{x} = (x_1, \ldots, x_t)$ be the normalized observation signal vector with t samples. The objective is to find a matrix $\hat{\mathbf{S}}$ whose rows are the different source signals. This matrix $\hat{\mathbf{S}}$ is an approximation of $\mathbf{S}$. In this article, we propose using the singular spectrum analysis [43] to decompose the observation in its principal components and later reconstruct both the drone and the source.

Singular spectrum analysis can be decomposed into two stages with two steps each. In the first stage, the observation is decomposed into its principal components. This process is done in two steps: embedding (section III-B1) and singular value decomposition (section III-B2). In the second stage, the target signals are reconstructed from groups of these principal components. This stage can be further divided into the following steps: grouping (section III-B3) and diagonal averaging (section III-B4).

As many principal components as the window length (denoted as the scalar k in this article) will be extracted from the observation. This window length is a chosen parameter that must fulfill the condition $1 < k \leq t$. We will discuss the method used to choose the value of this parameter in the results section, more specifically in the subsection IV-A2.

1) EMBEDDING

The embedding steps consists of constructing the trajectory matrix $\bar{\mathbf{X}}$ from the observation vector $\mathbf{x}$. This is the matrix whose $t-k+1$ columns are k samples lagged vectors of $\mathbf{x}$. $\bar{\mathbf{X}}$ is also a Hankel matrix because all the elements of its antidiagonals are equal. To construct this trajectory matrix, a hankelization function $\mathcal{H}_k$ that maps a vector to a Hankel matrix is defined in (3).

$$\bar{\mathbf{X}} = \mathcal{H}_k(\mathbf{x}) = \begin{bmatrix} x_1 & x_2 & \cdots & x_{t-k+1} \\ x_2 & x_3 & \cdots & x_{t-k+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_k & x_{k+1} & \cdots & x_t \end{bmatrix} \tag{3}$$

2) SINGULAR VALUE DECOMPOSITION

In this step, the lag-covariance matrix $\mathbf{C}$ of $\mathbf{x}$ is approximated from $\bar{\mathbf{X}}$ using (4). Note that, since $\mathbf{x}$ is normalized, $\mathbf{C}$ is also an approximation of the lag-correlation matrix. Furthermore, this is a Toeplitz matrix because all the elements of the diagonals are equal.

$$\mathbf{C} = \frac{\bar{\mathbf{X}}\bar{\mathbf{X}}^\intercal}{t} \tag{4}$$

This lag-covariance matrix is eigendecomposed by finding a matrix Λ (see (5)) with $\lambda_i > \lambda_{i+1}$ and a matrix $\mathbf{Q}$ (see (6)) with $||\mathbf{q}_i|| = 1$ such that they satisfy (7). This way, the values $\lambda_i \ldots \lambda_k$ in the diagonal of Λ are the eigenvalues of $\mathbf{C}$ sorted in descending order, and the columns $\mathbf{q}_1 \ldots \mathbf{q}_k$ of the matrix $\mathbf{Q}$ are the unitary eigenvectors of $\mathbf{C}$.

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_k \end{bmatrix} \tag{5}$$

$$\mathbf{Q} = \begin{bmatrix} | & | & & | & | \\ \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_{k-1} & \mathbf{q}_k \\ | & | & & | & | \end{bmatrix} \tag{6}$$

$$\mathbf{C} = \mathbf{Q}\Lambda\mathbf{Q}^{-1} \tag{7}$$

The principal components $\mathbf{p}_i$, rows of matrix $\mathbf{P}$, are given by the scalar projection of each of the k-samples lagged vectors of $\mathbf{x}$ over the eigenvector $\mathbf{q}_i$ as detailed in (8). Note that since the eigenvectors are unitary vectors, $||\mathbf{q}_i|| = 1$ and the projection is simplified to a single inner product. Additionally, the calculus of $\mathbf{P}$ can be further simplified into a single matrix product, as detailed in (9).

$$p_{i,j} = \frac{\langle \mathbf{q}_i, \bar{\mathbf{x}}_j^\intercal \rangle}{\|\mathbf{q}_i\|} = \langle \mathbf{q}_i, \bar{\mathbf{x}}_j^\intercal \rangle \tag{8}$$

$$\mathbf{P} = \bar{\mathbf{X}}^\intercal \mathbf{Q} \tag{9}$$

The principal components $\mathbf{p}_i$ represent temporal trends in the observation (e.g., if we had 3 components, one of the components could be related to the seasonality, another to an ascending trend, and the last one to noise). However, these components cannot be compared with the observation since they do not share the same amplitude scale.

The vector projections matrix $\bar{\mathbf{R}}_i$ of the lagged vectors over the $\mathbf{q}_i$ eigenvector are the trajectory matrix of the reconstructed components. That is, $\mathcal{H}^{-1}(\bar{\mathbf{R}}_i)$ is the reconstruction of the principal component $\mathbf{p}_i$ on the same domain as the observation. The trajectory matrix of the reconstructed components can be obtained from the principal components as explained in (10) or, in a vectorized way, as explained in (11).

$$\bar{\mathbf{r}}_{i,j} = \mathbf{q}_i p_{i,j} \tag{10}$$

$$\bar{\mathbf{R}}_i = \mathbf{q}_i \otimes \mathbf{p}_i^\intercal \tag{11}$$

3) GROUPING

These trajectory matrices of the reconstructed components must be grouped in as many disjoint sets as sources in the observation. In this particular case, since there are only a drone and an unknown source on the observation, the trajectory matrices of the reconstructed components are split into two different groups: the ones corresponding to the drone and the ones corresponding to the unknown source.

The first trajectory matrix, which has the strongest temporal trend, is associated with the source. The rest of the matrices are associated with the drone. This choice is validated in the results section, more specifically in subsection IV-A1. The trajectory matrix associated with each of the sources to separate is obtained by adding all the trajectory matrices associated with their corresponding disjoint set. In this article, the trajectory matrix of the source $\bar{\mathbf{R}}_{\text{source}}$ and the drone $\bar{\mathbf{R}}_{\text{drone}}$ are defined as detailed in (12).

$$\begin{aligned} \bar{\mathbf{R}}_{\text{source}} &= \bar{\mathbf{R}}_1 \\ \bar{\mathbf{R}}_{\text{drone}} &= \sum_{i=2}^{k} \bar{\mathbf{R}}_i \end{aligned} \tag{12}$$

4) DIAGONAL AVERAGING

The last step consists of transforming the trajectory matrices associated with the sources into the time series of each signal. To achieve this, the inverse of the hankelization function $\mathcal{H}_k^{-1}$ must be applied to each one of them as described in (13). In this article, in the same vein as the works [43], [46] the inverse of the hankelization function is defined as the function that matches each Hankel matrix to a vector whose elements are the average of each of the diagonals of the Hankel matrix.

$$\mathbf{r}_i = \mathcal{H}_k^{-1}(\bar{\mathbf{R}}_i) \tag{13}$$

Finally, the predictions of the source and the drone are the time series associated with the trajectory matrices $\bar{\mathbf{R}}_{\text{source}}$ and $\bar{\mathbf{R}}_{\text{drone}}$ found in the grouping (subsection III-B3) step (see (14)). Note that since the observation was normalized, these two predictions should be denormalized to match its scale. A summary of the algorithm described in this section can be found in the Algorithm 1.

$$\begin{aligned} \hat{\mathbf{s}}_{\text{source}} &= \mathbf{r}_{\text{source}} \\ \hat{\mathbf{s}}_{\text{drone}} &= \mathbf{r}_{\text{drone}} \end{aligned} \tag{14}$$

Algorithm 1 Singular Spectrum Analysis Algorithm.

```
Input: The observation ($\mathbf{x}$) and the window length ($k$).
Output: The estimated drone signal $\hat{\mathbf{s}}_{\text{drone}}$ and the
        estimated source signal $\hat{\mathbf{s}}_{\text{source}}$.
$t \leftarrow \text{len}(\mathbf{x})$;
$\bar{\mathbf{X}} = \mathcal{H}_k(\mathbf{x})$;
$\mathbf{C} \leftarrow (\bar{\mathbf{X}} \times \bar{\mathbf{X}}^\intercal)/t$;
$\Lambda, \mathbf{Q} \leftarrow \text{eigendecompose}(\mathbf{C})$;
$\mathbf{P} \leftarrow \bar{\mathbf{X}}^\intercal \times \mathbf{Q}$;
$₦ \leftarrow \mathsf{0}_{k \times k \times t}$;
for $i$ in range($k$) do
    $₦[i] \leftarrow \mathbf{Q}[:, i] \otimes \mathbf{P}[i, :]$;
end for
$\bar{\mathbf{R}}_{\text{source}} \leftarrow ₦[0]$;
$\bar{\mathbf{R}}_{\text{drone}} \leftarrow \mathbf{0}_{k \times t}$;
for $i$ in range($1, k$) do
    $\bar{\mathbf{R}}_{\text{drone}} \leftarrow \bar{\mathbf{R}}_{\text{drone}} + ₦[i]$;
end for
$\hat{\mathbf{s}}_{\text{source}} \leftarrow \mathcal{H}_k^{-1}(\bar{\mathbf{R}}_{\text{source}})$;
$\hat{\mathbf{s}}_{\text{drone}} \leftarrow \mathcal{H}_k^{-1}(\bar{\mathbf{R}}_{\text{drone}})$;
return $\hat{\mathbf{s}}_{\text{drone}}$, $\hat{\mathbf{s}}_{\text{source}}$;
```

C. OPTIMIZATION

The method previously explained has a $\mathcal{O}\left(tk^2\right)$ memory complexity and a $\mathcal{O}(tk^2)$ time complexity. In this section, we propose an optimization of said method with a $\mathcal{O}\left(tn + k^2\right)$ memory complexity and a $\mathcal{O}\left(tk + k^3\right)$ time complexity. Since usually $tn \gg k^2$, memory complexity is almost exclusively dependent on t and n in most situations ($\mathcal{O}\left(tn + k^2\right) \approx \mathcal{O}\left(tn\right)$). This optimization enables the method to use absurdly big window lengths and process signals with an increased length.

Korobeynikov [47] achieved a complexity of $\mathcal{O}(tk \log t)$ by using clever techniques based on the properties of Hankel matrice and Fourier transforms when performing matrix products. In this work, our approach achieves comparable results by replacing Hankel matrices operations with discrete convolution-based ones.

Our optimization process was focused around the hankelization function ($\mathcal{H}_k$). This function converts a t elements vector into a $k \times t - k + 1$ matrix. Since all the elements in the antidiagonals of a Hankel matrix are equal, they are uniquely determined by their first column and their last row. In other words, only t out of $k \cdot (t - k + 1)$ elements are relevant. By removing redundant elements, the amount of memory used can be heavily reduced.

As explained in the section III-B section, Singular Spectrum Analysis has two phases: decomposition of the signal in its principal components and reconstruction of the source signals from these components. The first of the phases was performed in two steps, embedding, and singular value decomposition. The second of the phases was, also, performed in two steps: grouping and diagonal averaging. In this optimized version, the embedding, singular value decomposition, and diagonal averaging steps are merged into a single one. Both steps of the two phases are merged into a single step. The new decomposition phase will be explained in more detail in section III-C1, and the new reconstruction function will be described in section III-C3.

1) EMBEDDING AND SINGULAR VALUE DECOMPOSITION

The objective of the embedding step was mostly to construct the trajectory matrix of the time signal. Later, in the second step, this trajectory matrix was used to estimate the lag-covariance matrix using (4). Note that, in that context, $\bar{\mathbf{X}}$ is a $k \times t - k + 1$ matrix. The time complexity of these two operations combined is $\mathcal{O}\left(tk^2\right)$ and the space complexity is $\mathcal{O}\left(tk\right)$. These two operations can be optimized using the observation in the next paragraph.

Let $\mathbf{a}$ be a row vector with a elements, $\mathcal{H}_b$ the hankelization function using a window length of b, $\mathbf{J}_x$ the $x \times x$ exchange matrix, and $\mathbf{0}_x$ the $x \times x$ zero matrix. It can be observed that the matrix product of the Hankel matrix of $\mathbf{a}$ by the transpose of itself is equivalent to constructing the cyclic matrix (denoted as the function $\mathcal{C}$) of the convolution of the first $a - b + 1$ elements of $\mathbf{a}$ in inverse order over itself as summarized in (15).

$$\mathcal{H}_b(\mathbf{a})\mathcal{H}_b(\mathbf{a})^\intercal = \mathcal{C}\left(\mathbf{a} * \left(\left(\mathbf{J}_{a-b+1}|\mathbf{0}_{b-1}\right)\mathbf{a}^\intercal\right)^\intercal\right) \tag{15}$$

With the help of the observation in (15), (4) can be rewritten as (16). Note that $\mathbf{x}$ is a t-elements vector, and that the matrix $(\mathbf{J}_{t-k+1}|\mathbf{0}_{k-1})$ does not have to be constructed in practice, as it is mathematical notation to denote the extraction of the first $t - k + 1$ elements of $\mathbf{x}$. Taking these two ideas into account, the time complexity is reduced to $\mathcal{O}\left(tk\right)$, and the space complexity to $\mathcal{O}\left(t + k^2\right)$.

$$\mathbf{C} = \frac{\mathcal{C}\left(\mathbf{x} * \left(\left(\mathbf{J}_{t-k+1}|\mathbf{0}_{k-1}\right)\mathbf{x}^\intercal\right)^\intercal\right)}{t} \tag{16}$$

The eigendecomposition of this matrix $\mathbf{C}$ (see (5), (6), and (7)) is the biggest bottleneck in the algorithm. The complexity of most eigendecomposition algorithms for square Hermitian and symmetric matrices have a computational complexity of $\mathcal{O}(n^3)$, where n is the size of the matrix.

2) PRINCIPAL COMPONENTS EXTRACTION AND DIAGONAL AVERAGING

The projection of the original signal over the eigenvectors performed in (8) and (9)has a $\mathcal{O}(tk^2)$ time complexity and a $\mathcal{O}(tk)$ space complexity in the original algorithm. This can be heavily improved by using the observation in (17) where $\mathbf{a}$ is a row vector with a elements, $\mathbf{b}$ is a column vector with b elements, $\mathcal{H}_b$ the hankelization function using a window length of b and $\mathbf{J}_x$ the $x \times x$ exchange matrix. That is, the product of the Hankel matrix of $\mathbf{a}$ by a vector $\mathbf{b}$ is equivalent to the convolution of the vector $\mathbf{b}$ in inverse order over $\mathbf{a}$.

$$\mathcal{H}_b(\mathbf{a})^\intercal\mathbf{b} = \mathbf{a} * (\mathbf{J}_b\mathbf{b})^\intercal \tag{17}$$

Using this observation, (9) can be rewritten as (18). The time complexity of this alternative is $\mathcal{O}(tk)$ for each component $\mathbf{p}_i$, that is a $\mathcal{O}(tk^2)$ complexity for the k elements. The space complexity, since only one component has to be stored at a given time, is $\mathcal{O}(t)$.

$$\mathbf{p}_i = \mathbf{x} * (\mathbf{J}_b \mathbf{q}_i)^\intercal \tag{18}$$

The matrices of vector projections $\bar{\mathbf{R}}_i$ were calculated from these $\mathbf{p}_i$ values using (10) and (11). These matrices are the trajectory matrices of the principal components of the signal, and diagonal averaging is then needed to retrieve them. The time and space complexity of this whole process was $\mathcal{O}(tk)$ for each reconstructed component.

This last step is optimized using the observation that follows. Let $\mathbf{a}$ be a row vector with a elements, $\mathbf{b}$ a column vector with b elements, $\mathbf{1}_n$ a vector of ones with n elements, $\mathcal{H}_b^{-1}$ the inverse of the hankelization function with a window length of b by diagonal averaging, and $\circledast$ the full discrete convolution operator defined as $\mathbf{v} \circledast \mathbf{w} = (\mathbf{0}_{w-1} \mid \mathbf{v} \mid \mathbf{0}_{w-1}) * \mathbf{w}$. Then, as described in (19), the inverse hankelization of the outer product of a column vector and a row vector is equivalent to the full discrete convolution $\mathbf{b}$ over $\mathbf{a}$ over the full discrete convolution of a vector of b ones over a vector of a ones. Moreover, the denominator of this fraction can be approximated as b.

$$\mathcal{H}_b^{-1}(\mathbf{b} \otimes \mathbf{a}) = \frac{\mathbf{a} \circledast \mathbf{b}^\intercal}{\mathbf{1}_a \circledast \mathbf{1}_b} \approx \frac{\mathbf{a} \circledast \mathbf{b}^\intercal}{b} \tag{19}$$

With this observation, (11) and (13) can be merged and rewritten as (20). The temporal complexity of the new method is $\mathcal{O}(tk)$ for each reconstructed component, and the spatial complexity is $\mathcal{O}(t)$.

$$\mathbf{r}_i = \frac{\mathbf{p}_i \circledast \mathbf{q}_i^\intercal}{\mathbf{1}_t \circledast \mathbf{1}_k} \approx \frac{\mathbf{p}_i \circledast \mathbf{q}_i^\intercal}{k} \tag{20}$$

3) GROUPING

In the unoptimized algorithm, the grouping was performed before dehankelizing the components using (12). In the optimized version, however, this grouping is done after dehankelizing them, as $\mathcal{H}_k^{-1}(\mathbf{C} + \mathbf{D}) = \mathcal{H}_k^{-1}(\mathbf{C}) + \mathcal{H}_k^{-1}(\mathbf{D})$ where $\mathbf{C}$ and $\mathbf{D}$ are two matrices of the same size. A summary of the optimized algorithm can be found in the Algorithm 2.

IV. EXPERIMENTAL RESULTS AND PERFORMANCE EVALUATION

In this section, we present the experiments carried out to validate the accuracy and performance of the proposal. In the section IV-A, the accuracy of the algorithm is validated by comparing it with that of related methods (subsection IV-A3), and justifying the selection of the parameters: window length (subsection IV-A2) and the components used to reconstruct the sources (subsection IV-A1). Furthermore, in the section IV-B, the execution time and memory consumption of the optimized algorithm are compared to the unoptimized algorithm.

Algorithm 2 Optimized Singular Spectrum Analysis Algorithm.

```
Input: The observation ($\mathbf{x}$) and the window length ($k$).
Output: The estimated drone signal $\hat{\mathbf{s}}_{\text{drone}}$ and the
        estimated source signal $\hat{\mathbf{s}}_{\text{source}}$.
$t \leftarrow \text{len}(\mathbf{x})$;
$\mathbf{C} \leftarrow \mathcal{C}(\mathbf{x} * \mathbf{x}[t-k :: -1])/t$;
$\Lambda, \mathbf{Q} \leftarrow \text{eigendecompose}(\mathbf{C})$;
$\mathbf{p} = \mathbf{x} * \mathbf{Q}[:: -1, 0]$;
$\hat{\mathbf{s}}_{\text{source}} \leftarrow (\mathbf{p} \circledast \mathbf{Q}[:, 0])/k$;
$\hat{\mathbf{s}}_{\text{drone}} \leftarrow \mathbf{0}_t$;
for $i$ in range$(1, k)$ do
    $\mathbf{p} \leftarrow \mathbf{x} * \mathbf{Q}[:: -1, i]$;
    $\hat{\mathbf{s}}_{\text{drone}} \leftarrow \hat{\mathbf{s}}_{\text{drone}} + (\mathbf{p} \circledast \mathbf{Q}[:, i])/k$;
end for
return $\hat{\mathbf{s}}_{\text{drone}}$, $\hat{\mathbf{s}}_{\text{source}}$;
```

A. SOURCE SEPARATION

The dataset used for validation in this work contains sounds recorded in the anechoic chamber shown in Fig. 4. To this end, a single microphone with a 51.2kHz sample rate was used. Three kinds of sounds were recorded: sounds emitted by the drone flying with different motor speeds, sounds emitted by a source, and a mix of the two. In the validation process, we created synthetic mixes using the expression (21). Once such example is portrayed in Fig. 3. Using these synthesized signals, a prediction is made, as exemplified in Fig. 2. This prediction example was the result of applying our method to the signal in Fig. 3, with a window length of 9 samples.

Using this dataset, three experiments have been carried out: the first of them is concerned with validating the distribution of the components used to reconstruct the source and the drone, the second of them studies the performance of the algorithm for different window lengths, and the last of them compares our optimized method with the naive implementation of singular spectrum analysis.

1) INFLUENCE OF THE COMPONENTS USED TO RECONSTRUCT THE SOURCE ON THE ERROR

To find the best number of components used to reconstruct the source, we used several executions of the algorithm with different parameter values: the window length, the relative power of the source sound with respect to the drone sound, and the number of components used to reconstruct the source. For the first of the parameters, the window length, we used 24 different values equispaced between 2 and 25 samples. For the power of the source, 50 values equispaced between −5db and 5db were used.

The number of components used to reconstruct the source ranged from one to the number of components extracted by means of the singular spectrum analysis (i.e., as many as the window length). For a value of n components, the first n components were used to reconstruct the source.

The mixes used to perform this experiment have been generated synthetically, using drone sounds and source sounds from the dataset. The mixing system is the one described

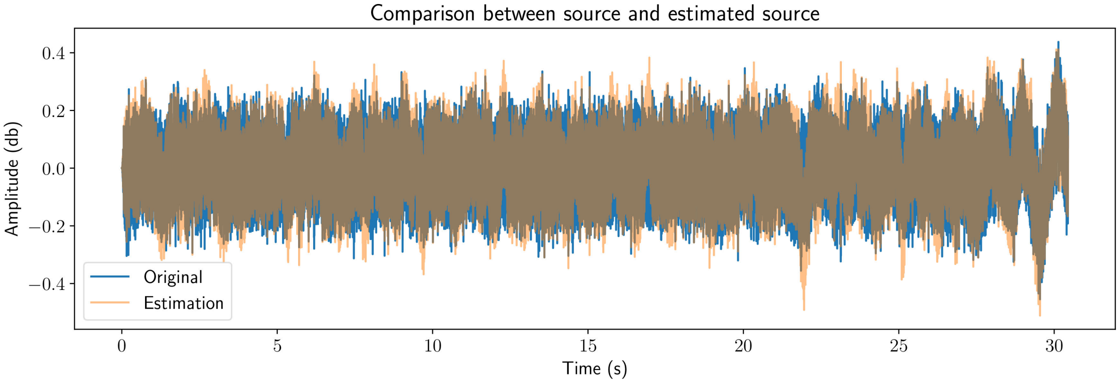

FIGURE 2. Example of source signal prediction.

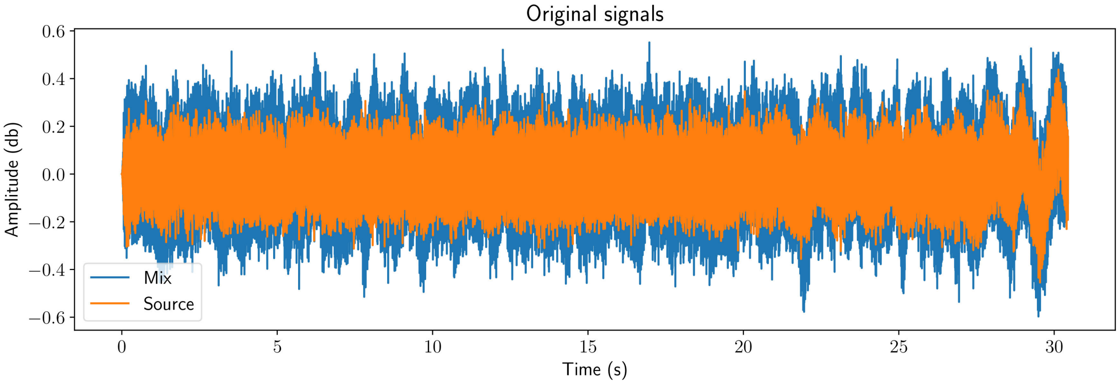

FIGURE 3. Example of source and mix signals.

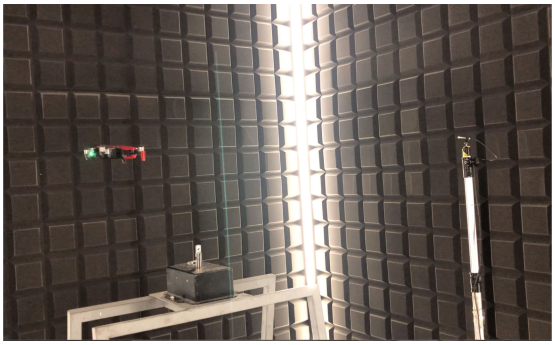

FIGURE 4. Recording the sound of a drone in the anechoic chamber.

in (21), where d is the relative power of the source with respect to the drone in decibels.

$$\mathbf{x}^d = \frac{1}{1+10^{\frac{d}{20}}} \cdot \mathbf{s}_{\text{drone}} + \frac{10^{\frac{d}{20}}}{1+10^{\frac{d}{20}}} \cdot \mathbf{s}_{\text{source}} \tag{21}$$

All the possible combinations of the parameters have been tested. In all of them, the error has been measured using the root-mean-square error normalized by the mean of the reference signal (NRMSE).

The results of this experiment are summarized in Fig. 5. There, the average NRMSE for every relative power is shown using a color scale versus the window length (horizontal axis) and the number of components used to reconstruct the source (vertical axis). Note that the top half of the representation is left blank on purpose, as it is not possible to use more components to reconstruct the source than the number of components extracted, which are as many as the window length. In this figure it becomes apparent that the optimum number of components to reconstruct the source with the smallest error possible is equal to 1, regardless of the window length.

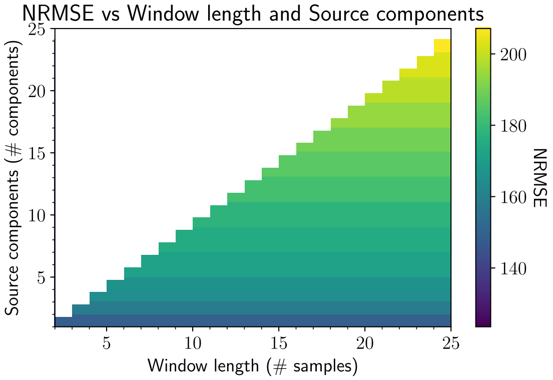

FIGURE 5. Error versus the number of components used to reconstruct the source and the window length.

As a result of this experiment, since using more than a single component to reconstruct the source has a negative impact on the result, only the first component is used to

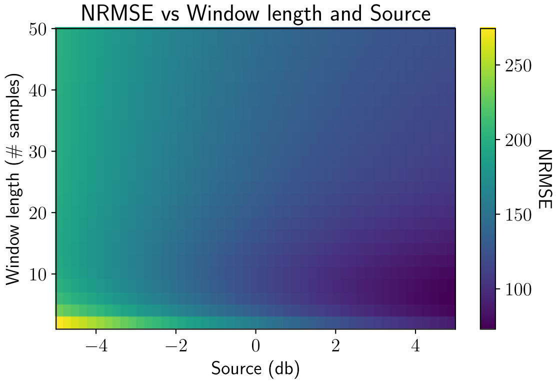

FIGURE 6. Error versus the number of components and source relative power.

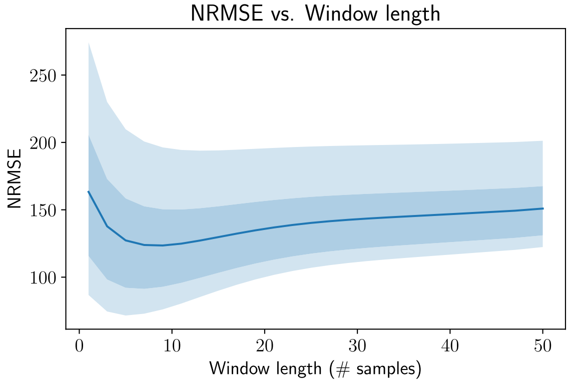

FIGURE 7. Error versus number of components.

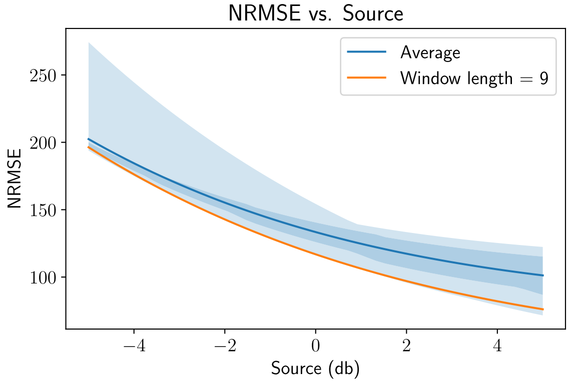

FIGURE 8. Error versus source power.

estimate the source in every other experiment and real usage scenarios.

2) INFLUENCE OF THE WINDOW LENGTH IN THE ERROR

In the work [46], the authors recommend using a window length equal to the number of sources to estimate. However, in our work, we have chosen to conduct an exhaustive study with a lot of different configurations to check if this value is, in fact, the most appropriate for our case study. To achieve this, we executed the algorithm several times with different window lengths and relative powers of the source with respect to the drone.

The window length took 25 equispaced values between 1 sample and 25 samples. The relative power took 50 equispaced values between −5db and 5db, as in the experiment above. Again, in this experiment, drone and source sounds were mixed using (21). A total of 1250 executions were performed and the error was once again estimated using NRMSE.

The results of this experiment are portrayed in Fig. 6, Fig. 7, and Fig. 8. The visualization in Fig. 6 is a heat map that shows how the error varies with respect to the window length and the power of the source. In a more specific way, Fig. 7 and Fig. 8 show the evolution of the NRMSE with reference to the window length and the power of the source. In these two, the dark blue line corresponds to the average error, the darker blue zone shows the interquartile range, and the lighter blue zone shows the range between the minimum and the maximum for that parameter value.

The orange line is the evolution of the error versus the relative power of the source using the window length that achieves the lowest average error in Fig. 7 (this value is considered the optimal window length for this use case). Based on these visualizations, it can be observed that the value of the window length that minimizes the average NRMSE is 9 samples.

3) COMPARISON

In this section, we compare the method used in this article to two other relevant methods, namely [26] and [28]. All three algorithms were applied on different mixes of drone and source sounds to predict the source. These mixes were generated using 21 and different relative powers between the source with respect to the drone, ranging from −5db and 5db. The NRMSE score is used to evaluate the predictions of the algorithm. Finally, the scores of the predictions of each algorithm are averaged. The average scores are listed in Table 2. In this table, it can be seen that, on average, the approach put forward in this article is more efficient than the other two.

TABLE 2. Comparison of the method selected with other relevant methods.

Method	Average NRMSE
Singular spectrum analysis (method used)	123.88
Wavelet transform denoising [28]	134.21
Second wavelet transform denoising [26]	163.30

B. OPTIMIZATION

Both in its optimized and unoptimized version, the algorithm was implemented using the *Python 3* language and the libraries *NumPy* (https://numpy.org/) and *SciPy* (https://www.scipy.org/).

In both of these implementations, two metrics were evaluated: consumed memory (in megabytes) and execution time (in seconds). With the help of the library *memory_profiler*

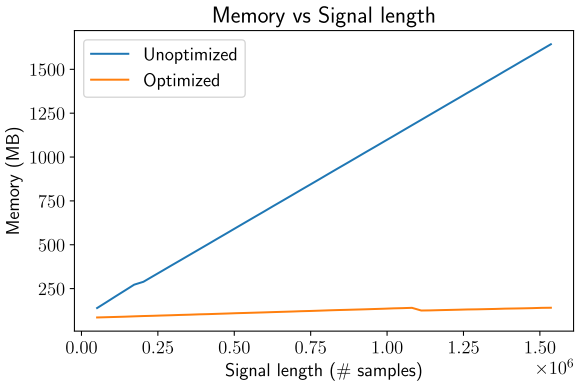

FIGURE 9. **Memory consumed versus signal length.**

(https://pypi.org/project/memory-profiler/), the memory consumed by the algorithm was retrieved at regular time intervals during its execution. The selected value for the consumed memory metric is the maximum value registered with *memory_profiler* because this will be the minimum memory the algorithm will need. The value of the execution time metric has been measured using the *timeit* module from the standard library of *Python*.

The effect of two parameters (the signal length, and the window length) on these metrics was studied as well. For the signal length, 50 equispaced values between $1 \cdot 51200$ samples and $30 \cdot 51200$ samples were used. The choice of these limits is not arbitrary, $1 \cdot 51200$ is the number of samples in a one-second audio file recorded with a 51.2kHz microphone, and, in the same way, $30 \cdot 51200$ is the number of samples in a 30 seconds-long recording. For the window length, 29 integer values equispaced between 2 and 30 samples were chosen.

Considering these metrics and parameters, four experiments were carried out. In the first one of them, the consumed memory is evaluated with respect to the signal length on both algorithms (the window length is kept fixed at a value of 10). In the second, the consumed memory is evaluated with respect to the window length on both algorithms (the signal length is kept fixed at $5 \cdot 51200$ samples). In the third, the execution time is evaluated with respect to the signal length on both algorithms (the window length is kept fixed at a value of 10). In the last, the execution time is evaluated with respect to the window length of both algorithms (the signal length is kept fixed at $5 \cdot 51200$ samples).

To ensure that the results of these experiments are as close to reality as possible, the experiments were repeated 10 times on the same machine, using different random values as input for the algorithm. Moreover, in order to avoid noise produced by the algorithm of the garbage collector of *Python*, each execution was carried on a different instance of the interpreter.

The results of the 10 executions of each of the experiments can be seen in Fig. 9, Fig. 10, Fig. 11, and Fig. 12. Similarly to the previous sections, these plots have a darker line associated with the median, a slightly lighter zone corresponding to the interquartile range, and an even lighter zone between the minimum and the maximum.

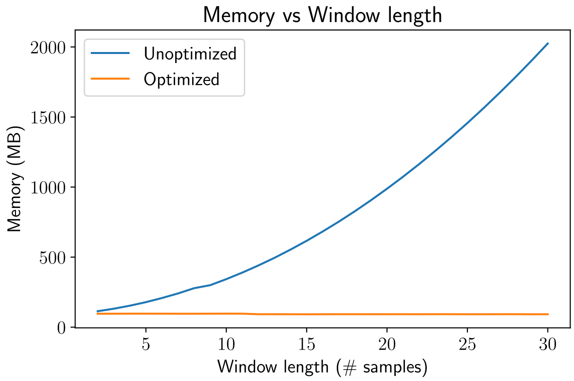

FIGURE 10. **Memory consumed versus window length.**

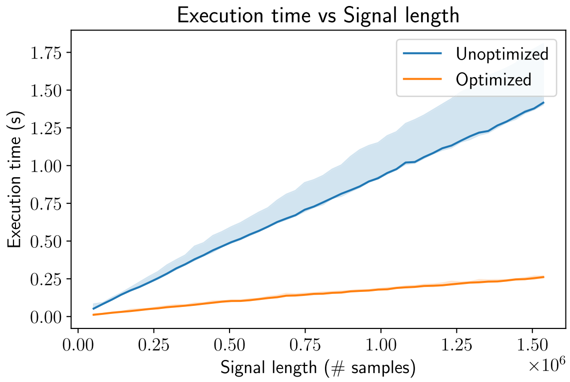

FIGURE 11. **Time elapsed versus signal length.**

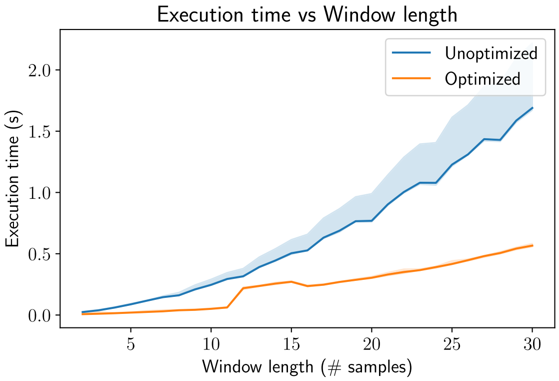

FIGURE 12. **Time elapsed versus window length.**

Note that both lines in the plot of the four experiments use this representation scheme. However, only the lines corresponding to the unoptimized version show deviation, and that only on the execution time. This deviation is caused by a few executions that take more time than the others, as the interquartile range is very close to the median. We hypothesize that the maximum execution time is caused by the first execution, while all the other executions are close to the median. This first execution would be interpreted directly from the source code by the *Python* runtime, while the others would execute as bytecode cached after the first execution. This hypothesis is reinforced by Fig. 13, where the execution times of 10 different tests carried with a $30 \cdot 51200$ sample signal length and a 10-sample window length are depicted.

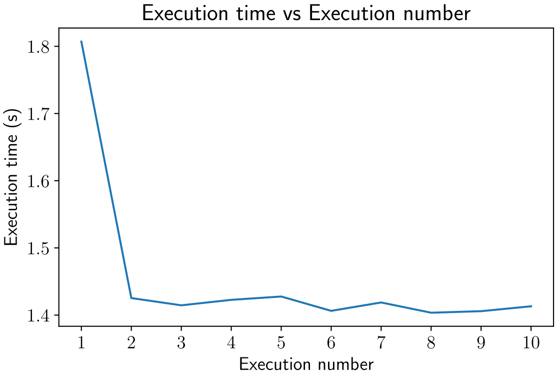

FIGURE 13. Execution time for the 10 different tests.

In the other cases, the deviation with respect to the median is not significant compared to the scale of the visualization.

V. CONCLUSION AND FUTURE WORK

Drone-noise reduction in single-channel recordings is a very challenging problem. One of the reasons is that, when approached as a blind source separation problem, the issue is ill-posed problem: since there are less observations (only one) than sources (two), the latter are not uniquely determined. As such, algorithms like beam-forming, independent component analysis, and principal component analysis are unsuitable for this problem. More so, since drone-noise is broadband in nature and its frequencies of emission usually overlap with the frequencies of other sounds; low-pass, band-pass, and high-pass filters are rendered unusable. Furthermore, drone-noise is highly non-stationary, so the Wiener filter is not applicable. Lastly, in our case of study, neural networks and genetic algorithms are also a bad approach because there was not enough data to prevent overfitting. As a result, the options available are greatly limited.

In our case study, which consists of a drone-based system for audio recording in inaccessible locations, we chose the singular spectrum analysis. We adjusted the parameters of our method using a grid search. Furthermore, we proposed an optimization of the algorithm with a spatial complexity of $\mathcal{O}(nt)$ (where t is the signal length and n is the number of sources to reconstruct). Several experiments were performed to validate the method, both accuracy-wise and performance-wise.

The proposed approach succeeded in separating the sound of the drone and the sound of the source with great efficiency. For example, in order to perform the experiment in section IV-A2, the algorithm was executed a total of 1250 times, using a signal with a length of $30 \cdot 51200$ samples and a window length that ranged from 1 to 50. All these executions took only 37 minutes and 29 seconds, that is an average 1.8 seconds per execution.

However, since the ego-noise cancellation problem was approached as a blind source separation task, the characteristics of the sound produced by the drone were ignored. Preliminary analyses shows that the noise produced by the drone could be easily predicted and dependent on the angular velocity of the motors. In future works, we want to study different algorithms that exploit the characteristics of the sound produced by the drone to achieve better results. Furthermore, we want to study how the velocity of the motors, the model of the motors, and the model of the propellers affect this noise. Last but not least, we want to find out how these parameters could be predicted from a single sound sample.

REFERENCES

[1] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, ‘‘Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges,’’ *IEEE Access*, vol. 7, pp. 48572–48634, 2019, doi: 10.1109/ACCESS.2019.2909530.

[2] B. Allred, L. Martinez, M. K. Fessehazion, G. Rouse, T. N. Williamson, D. Wishart, T. Koganti, R. Freeland, N. Eash, A. Batschelet, and R. Featheringill, ‘‘Overall results and key findings on the use of UAV visible-color, multispectral, and thermal infrared imagery to map agricultural drainage pipes,’’ *Agricult. Water Manage.*, vol. 232, Apr. 2020, Art. no. 106036, doi: 10.1016/j.agwat.2020.106036.

[3] L. A. Silva, H. S. S. Blas, D. P. García, A. S. Mendes, and G. V. González, ‘‘An architectural multi-agent system for a pavement monitoring system with pothole recognition in UAV images,’’ *Sensors*, vol. 20, no. 21, p. 6205, Oct. 2020, doi: 10.3390/s20216205.

[4] V. Hoehne and R. Luce, ‘‘The quieted aircraft as a military tool,’’ in *Proc. AIAA Aircr. Design Oper. Meeting*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, Jul. 1969, pp. 1–11, doi: 10.2514/6.1969-792.

[5] J. Cross and M. Watts, ‘‘In-flight acoustic testing techniques using the YO-3A acoustic research aircraft,’’ in *Proc. 2nd Flight Test. Conf.* Reston, VA, USA: American Institute of Aeronautics and Astronautics, Nov. 1983, pp. 2754 and 1–8, doi: 10.2514/6.1983-2754.

[6] T. Ohata, K. Nakamura, T. Mizumoto, T. Taiki, and K. Nakadai, ‘‘Improvement in outdoor sound source detection using a quadrotor-embedded microphone array,’’ in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 1902–1907, doi: 10.1109/IROS.2014.6942813.

[7] P. Alloza and B. Vonrhein, ‘‘Localización sonora en drones mediante una acoustic camera,’’ in *Proc. 51° Congreso Español de Acústica*, 2020, pp. 1–12. [Online]. Available: http://www.spacustica.pt/acustica2020/papers/ID99.pdf

[8] W. Fiebig and D. Dąbrowski, ‘‘Use of acoustic camera for noise sources localization and noise reduction in the industrial plant,’’ *Arch. Acoust.*, vol. 45, no. 1, pp. 111–117, 2020, doi: 10.24425/aoa.2020.132487.

[9] L. Wang and A. Cavallaro, ‘‘A blind source separation framework for ego-noise reduction on multi-rotor drones,’’ *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 2523–2537, 2020, doi: 10.1109/TASLP.2020.3015027.

[10] H.-G. Ma, Q.-B. Jiang, Z.-Q. Liu, G. Liu, and Z.-Y. Ma, ‘‘A novel blind source separation method for single-channel signal,’’ *Signal Process.*, vol. 90, no. 12, pp. 3232–3241, 2010, doi: 10.1016/j.sigpro.2010.05.029.

[11] B. Widrow, J. R. Glover, J. M. McCool, J. Kaunitz, C. S. Williams, R. H. Hearn, J. R. Zeidler, J. E. Dong, and R. C. Goodlin, ‘‘Adaptive noise cancelling: Principles and applications,’’ *Proc. IEEE*, vol. 63, no. 12, pp. 1692–1716, Dec. 1975, doi: 10.1109/PROC.1975.10036.

[12] B. Kang, H. Ahn, and H. Choo, ‘‘A software platform for noise reduction in sound sensor equipped drones,’’ *IEEE Sensors J.*, vol. 19, no. 21, pp. 10121–10130, Nov. 2019, doi: 10.1109/JSEN.2019.2927370.

[13] G. Sinibaldi and L. Marino, ‘‘Experimental analysis on the noise of propellers for small UAV,’’ *Appl. Acoust.*, vol. 74, no. 1, pp. 79–88, Jan. 2013, doi: 10.1016/j.apacoust.2012.06.011.

[14] D. B. Hanson, ‘‘Influence of propeller design parameters on far-field harmonic noise in forward flight,’’ *AIAA J.*, vol. 18, no. 11, pp. 1313–1319, Nov. 1980, doi: 10.2514/3.50887.

[15] R. Van Rompaey and M. Moonen, ‘‘GEVD based speech and noise correlation matrix estimation for multichannel Wiener filter based noise reduction,’’ in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 2544–2548, doi: 10.23919/EUSIPCO.2018.8553109.

[16] A. Shamsa, S. Ghorshi, and M. Joorabchi, ‘‘Noise reduction using multi-channel FIR warped Wiener filter,’’ in *Proc. 13th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Mar. 2016, pp. 531–536, doi: 10.1109/SSD.2016.7473687.

[17] J. G. Lopez and J. I. Marin-Hurtado, "Multichannel Wiener filter method for noise reduction in reverberant environments for binaural hearing aids," in *Proc. 20th Symp. Signal Process., Images Comput. Vis. (STSIVA)*, Sep. 2015, pp. 1–5, doi: 10.1109/STSIVA.2015.7330392.

[18] A. Sarafnia and S. Ghorshi, "Noise reduction of speech signal using Bayesian state-space Kalman filter," in *Proc. 19th Asia–Pacific Conf. Commun. (APCC)*, Aug. 2013, pp. 545–549, doi: 10.1109/APCC.2013.6766008.

[19] H. Ozcan, S. Sahin, M. Mentesoglu, and F. Pir, "Indoor reduction of noise in RF signal with Kalman filter," in *Proc. 23nd Signal Process. Commun. Appl. Conf. (SIU)*, May 2015, pp. 2013–2016, doi: 10.1109/SIU.2015.7130260.

[20] J. Valin, "A hybrid DSP/deep learning approach to real-time full-band speech enhancement," in *Proc. IEEE 20th Int. Workshop Multimedia Signal Process. (MMSP)*, Aug. 2018, pp. 1–5, doi: 10.1109/MMSP.2018.8547084.

[21] L. Zhang, M. Wang, Q. Zhang, and M. Liu, "Environmental attention-guided branchy neural network for speech enhancement," *Appl. Sci.*, vol. 10, no. 3, p. 1167, Feb. 2020, doi: 10.3390/app10031167.

[22] L. Wang and A. Cavallaro, "Deep learning assisted time-frequency processing for speech enhancement on drones," *IEEE Trans. Emerg. Topics Comput. Intell.*, early access, Aug. 24, 2020, doi: 10.1109/TETCI.2020.3014934.

[23] I. Hermawan, G. Jati, D. M. S. Arsa, and W. Jatmiko, "Particle swarm optimization based nonlocal means for denoising ECG signal," in *Proc. IEEE Int. Conf. Signals Syst. (ICSigSys)*, Jul. 2019, pp. 69–73, doi: 10.1109/ICSIGSYS.2019.8811084.

[24] C.-C. Liu, S.-T. Hsieh, and W.-C. Huang, "Evolutionary noise reduction based on correlation evaluation," in *Proc. IET Int. Conf. Frontier Comput. Theory, Technol. Appl.*, 2010, pp. 411–415, doi: 10.1049/cp.2010.0597.

[25] D. Yan, L. Zhang, L. Zhang, and Y. Wang, "A reconfigurable 1/2/3-order butterworth LP/CBP filter with noise canceller for sub-GHz applications," in *Proc. 13th IEEE Int. Conf. Solid-State Integr. Circuit Technol. (ICSICT)*, Oct. 2016, pp. 1330–1332, doi: 10.1109/ICSICT.2016.7998730.

[26] F. Yan and Z. Chunqin, "Seismic data de-noising based on second wavelet transform," in *Proc. Int. Conf. Adv. Comput. Theory Eng.*, Dec. 2008, pp. 186–189, doi: 10.1109/ICACTE.2008.118.

[27] Z. Shan, J. Zhou, and J. Chen, "Background noise suppression of magnetic anomaly signal based on wavelet transform," in *Proc. IEEE 17th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2017, pp. 333–337, doi: 10.1109/ICCT.2017.8359656.

[28] C.-H. Chang, T.-M. Wang, and H.-L. Hsu, "Denoising of mixed noises in ECG with separate noise estimators based on discrete wavelet transform," in *Proc. Int. Conf. Adv. Mater. Sci. Eng. (ICAMSE)*, Nov. 2016, pp. 562–564, doi: 10.1109/ICAMSE.2016.7840200.

[29] B. Zhao, J.-A. Yang, and M. Zhang, "Research on blind source separation and blind beamforming," in *Proc. Int. Conf. Mach. Learn. Cybern.*, vol. 7, 2005, pp. 4389–4393, doi: 10.1109/ICMLC.2005.1527711.

[30] H. Sawada, S. Araki, R. Mukai, and S. Makino, "Grouping separated frequency components by estimating propagation model parameters in frequency-domain blind source separation," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 15, no. 5, pp. 1592–1604, Jul. 2007, doi: 10.1109/TASL.2007.899218.

[31] X. Wang, S. Ou, Y. Gao, and X. Guo, "A new fast nonlinear principal component analysis algorithm for blind source separation," in *Proc. 12th Int. Conf. Fuzzy Syst. Knowl. Discovery (FSKD)*, Aug. 2015, pp. 1626–1630, doi: 10.1109/FSKD.2015.7382188.

[32] M.-C. Wu and K.-C. Chen, "Sparse PCA via hard thresholding for blind source separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2539–2543, doi: 10.1109/ICASSP.2016.7472135.

[33] T. Zeng, J. Ma, and M. Dong, "Environmental noise elimination of heart sound based on singular spectrum analysis," in *Proc. Cairo Int. Biomed. Eng. Conf. (CIBEC)*, Dec. 2014, pp. 158–161, doi: 10.1109/CIBEC.2014.7020943.

[34] Z. Sun, X. Wang, X. Wang, K. Sun, and Q. Tan, "Removal of baseline wander in ECG signals using singular spectrum analysis," in *Proc. IEEE 9th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2019, pp. 391–394, doi: 10.1109/ICEIEC.2019.8784679.

[35] B. V. Hemambhar and S. J. Rani, "Denoising of ECG signals using fuzzy based singular spectrum analysis," in *Proc. IEEE Recent Adv. Intell. Comput. Syst. (RAICS)*, Dec. 2018, pp. 1–5, doi: 10.1109/RAICS.2018.8635072.

[36] B. Zhou and Z. Liu, "Method of multi-resolution and effective singular value decomposition in under-determined blind source separation and its application to the fault diagnosis of roller bearing," in *Proc. 11th Int. Conf. Comput. Intell. Secur. (CIS)*, Dec. 2015, pp. 462–465, doi: 10.1109/CIS.2015.117.

[37] L.-Y. Gao, M.-Z. Liu, J.-Y. Yue, and Y.-H. Tian, "Source number estimation based on improved singular value decomposition at low SNR," in *Proc. IEEE 9th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2019, pp. 1–4, doi: 10.1109/ICEIEC.2019.8784699.

[38] Y. Wang, H. Chen, D. He, J. Peng, B. Yuan, and Y. Li, "Study on improved singular value decomposition de-noising method applied to UAV flight parameter data," in *Proc. IEEE 20th Int. Conf. High Perform. Switching Routing (HPSR)*, May 2019, pp. 1–6, doi: 10.1109/HPSR.2019.8808112.

[39] P. Misra, A. A. Kumar, P. Mohapatra, and P. Balamuralidhar, "Aerial drones with location-sensitive ears," *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 154–160, Jul. 2018, doi: 10.1109/MCOM.2018.1700775.

[40] A. Schmidt, A. Deleforge, and W. Kellermann, "Ego-noise reduction using a motor data-guided multichannel dictionary," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*. Daejon, South Korea: IEEE/RSJ, Oct. 2016, pp. 1281–1286, doi: 10.1109/IROS.2016.7759212.

[41] W. N. Alexander and J. Whelchel, "Flyover noise of multi-rotor sUAS," in *Proc. INTER-NOISE NOISE-CON Congr. Conf.*, vol. 259, no. 7. Institute of Noise Control Engineering, 2019, pp. 2548–2558.

[42] A. Zaknich, *Principles of Adaptive Filters and Self-learning Systems* (Advanced Textbooks in Control and Signal Processing). London, U.K.: Springer-Verlag. Accessed: 2005, doi: 10.1007/b138890.

[43] N. Golyandina and A. Zhigljavsky, *Singular Spectrum Analysis for Time Series*. Berlin, Germany: Springer, 2013, doi: 10.1007/978-3-642-34913-3.

[44] S. Makino, T.-W. Lee, and H. Sawada, *Blind Speech Separation* (Signals and Communication Technology). Amsterdam, The Netherlands: Springer, 2007, doi: 10.1007/978-1-4020-6479-1.

[45] L. A. Silva, V. R. Q. Leithardt, C. O. Rolim, G. V. González, C. F. R. Geyer, and J. S. Silva, "PRISER: Managing notification in multiples devices with data privacy support," *Sensors*, vol. 19, no. 14, p. 3098, Jul. 2019, doi: 10.3390/s19143098.

[46] R. Wang, H.-G. Ma, G.-Q. Liu, and D.-G. Zuo, "Selection of window length for singular spectrum analysis," *J. Franklin Inst.*, vol. 352, no. 4, pp. 1541–1560, Apr. 2015, doi: 10.1016/j.jfranklin.2015.01.011.

[47] A. Korobeynikov, "Computation- and space-efficient implementation of SSA," *Statist. Interface*, vol. 3, no. 3, pp. 357–368, 2010, doi: 10.4310/SII.2010.v3.n3.a9.

FRANCISCO GARCÍA ENCINAS received the bachelor's degree in computer engineering and the master's degree in intelligent systems from the University of Salamanca, Spain, in 2019 and 2020, respectively, where he is currently pursuing the Ph.D. degree. He currently works as a Researcher with the ESALab, Research Group, University of Salamanca. During the summer of 2018, he interned at Global Exchange as a full stack Web Developer. From late 2018 to mid-2019, he has worked as a Scientific Software Programmer with the ALF-USAL Research Group, University of Salamanca.

LUÍS AUGUSTO SILVA received the degree in Internet systems from the Federal Institute of Santa Catarina (IFC), Camboriú, Brazil, in February 2017, and the master's degree in applied computing from the University of Itajaí Valley, Brazil, in 2019. He is currently pursuing the Ph.D. degree in computer engineering with the Universidad de Salamanca, Spain. His research during his master's degree included the field of notification systems, the IoT, and data privacy. During the master's degree, he was a collaborating Researcher with the Laboratory of Embedded and Distributed Systems (LEDS), UNIVALI, collaborating with research projects related to the Internet of Things. Since August 2020, he has been a Researcher with the Expert Systems and Applications Laboratory (ESALab), Universidad de Salamanca. His research interests include the Internet of Things, embedded drone systems, and data privacy applied to smart environments.

ANDRÉ SALES MENDES received the degree in computer engineering and the master's degree in intelligent systems from the University of Salamanca. He is currently pursuing the Ph.D. degree in computer engineering, focusing on artificial intelligence and expert robots. He is currently a member of the Expert Systems and Applications Laboratory Research Group, University of Salamanca. In addition, he has several publications in JCR indexed journals.

GABRIEL VILLARRUBIA GONZÁLEZ received the master's degree in intelligent systems from the University of Salamanca, in 2012, the master's degree in Internet security in 2014, the master's degree in information systems management in 2015, and the Ph.D. degree from the Department of Computer Science and Automation, University of Salamanca. He was a Computer Engineer with the Pontifical University of Salamanca, in 2011. He is currently a Research Professor with the Department of Informatics. Throughout his training, he has followed a well-defined line of research, focused on the application of multi-agent systems to ambient intelligence environments, with special attention to the definition of intelligent architectures and the fusion of information.

VALDERI REIS QUIETINHO LEITHARDT (Member, IEEE) received the Ph.D. degree in computer science from INF-UFRGS, Brazil, in 2015. He is currently an Adjunct Professor with the Polytechnic Institute of Portalegre and a Researcher integrated into the VALORIZA Research Group, School of Technology and Management (ESTG). He is also a collaborating Researcher with the following research groups: COPELABS, Universidade Lusófona de Lisboa, Portugal; Telecommunications Institute of Portugal, IT Branch Covilhã, Portugal; Department of Informatics, University Beira Interior, Covilhã, Portugal; Laboratory of Embedded and Distributed Systems, University of Vale do Itajaí (UNIVALI), Brazil; and Expert Systems and Applications Laboratory, University of Salamanca, Spain. His research interests include distributed systems with a focus on data privacy, communication and programming protocols, involving scenarios and applications for the Internet of Things, smart cities, big data, cloud computing, and Blockchain.

JUAN FRANCISCO DE PAZ SANTANA received the degree in technical engineering in systems computer sciences in 2003, and the Engineering degree in computer sciences, the degree in statistics, and the Ph.D. degree in computer science from the University of Salamanca, Spain, in 2005, 2007, and 2010, respectively. He is currently a Full Professor with the University of Salamanca, where he is also a Researcher with the Expert Systems and Applications Laboratory (ESALab). He has been a coauthor of published articles in several journals, workshops, and symposiums.