Unattended Acoustic Sensor Systems For Source Detection, Classification, and Tracking

Vladimir Yaremenko, Mahmood R. Azimi-Sadjadi, *Life Member, IEEE*, and Jarrod Zacher

***Abstract*— Distributed and decentralized sensing offers a new and promising paradigm for surveillance, reconnaissance, and situation awareness. This paper introduces a new fully decentralized decision-making platform referred to as *environmental monitoring station* (EMS) for sensor-level detection, classification, and tracking of acoustic airborne sources in national parks. This custom-designed field-programmable gate array-based platform allows for near real-time transient source detection and classification using the 1/3 octave spectral data extracted locally from the streaming acoustic data. Source tracking through successive angle-of-arrival (AoA) estimation is also implemented locally on the EMS system using an array of microphones. The general headings of the sources generated using the AoA history and the source labels are transmitted to a park station via two possible wireless links. Field test results are provided to evaluate the performance of the overall system.**

***Index Terms*— Acoustic signal processing, array processing, embedded and field-programmable gate array (FPGA) systems, noise monitoring, source tracking, transient detection and classification.**

I. Introduction

SYSTEMS for acoustic transient signal detection, classification, and tracking could have many important applications including surveillance, situational awareness, and monitoring. The existing detection and classification algorithms are either too sophisticated and resource demanding to be implemented on the available unattended sensor systems or are too simplistic for most realistic applications; hence, rendering them of limited use for decentralized processing on sensor platforms. In addition, centralized processing and decision-making require large bandwidth for data transmission which in turn limits the ability to process acoustic data streams for real-time surveillance and monitoring applications. Thus, new solutions are needed to develop and design sensor platforms with adequate processing, communication, and power management capabilities together with the appropriate transient detection, classification, and tracking algorithms that can be implemented locally for intelligent decentralized processing of acoustic data streams.

Manuscript received January 9, 2018; revised April 6, 2018; accepted May 28, 2018. Date of publication July 23, 2018; date of current version December 24, 2018. This work was supported by the National Park Service through a Cooperative Agreement under Grant P14AC00728. The Associate Editor coordinating the review process was Yong Yan. *(Corresponding author: Mahmood R. Azimi-Sadjadi.)*

The authors are with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523 USA (e-mail: azimi@engr.colostate.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2018.2849458

Several systems have recently been developed and prototyped to provide different acoustic/sonar detection, classification, and/or tracking capabilities. The system in [1] was developed to track dolphin population and habitats using an array of hydrophones attached to the rear of a boat. The field-programmable gate array (FPGA) core in this system is mainly used for data acquisition and buffering while the bulk of the processing including wavelet-based filtering, detection, and bearing angle estimation is implemented on a laptop. A perceptron neural network computes the range and bearing angle associated with the detected underwater source. The system is not decentralized and lacks networking capabilities, power management, and source identification capabilities. Pianegiani *et al.* [2] proposed a three-tier decentralized low-power wireless sensor network solution for vehicle classification using their acoustic signatures. The first tier consists of clusters of wireless sensor nodes (e.g., Mica2 motes) sending the collected data to gateway cluster heads which in turn perform feature extraction and preliminary decision-making. The third tier is the base station integrated with other base stations via the Internet to perform high-level decision-making, decision fusion, and network management. The main goal of the design is to reduce the power consumption of wireless sensor networks to distribute the processing among cluster heads and the base stations. The system employs power detector, spectral features, and support vector machine for event classification. Although the system offers acoustic event classification similar to the environmental monitoring station (EMS), it is structurally and algorithmically different in many ways. It relies on deploying a multitude of clusters of wireless sensor nodes, gateways, and base stations communicating through wireless local area network and Internet access. The nodes do not use FPGA boards and lack many capabilities of the proposed system including microphone array processing and source trajectory estimation. Sallai *et al.* [3] designed an FPGA-based helmet-mounted sensor node for counter-sniper applications. The sensor board supports four microphone channels forming a small array for computing the angle of arrival (AoA) of the muzzle blast and ballistic shockwave wavefronts using the time-of-arrival estimates for each microphone. The fusion algorithm at a base station then estimates the shooter location, bullet trajectory, and caliber, as well as the weapon type using the received information from multiple nodes forming an *ad hoc* network. Although there are some similarities between their sensor node and the EMS architectures, their functionality and applications are different. Finally, William and Hoffman [4] and Akhtar *et al.* [5] developed

different neural network-based algorithms for the detection and classification of ground and airborne vehicles without providing any hardware structure for their implementation.

The EMS system proposed in this paper is designed and built to provide fully decentralized sensor-level acoustic source detection and classification. Each EMS system also supports multiple microphones forming an array allowing for sensor-level source AoA estimation and tracking. The temporal history of the AoA tracks would then generate approximate flight paths of the detected and classified sources relative to the array. This capability cannot be achieved using the existing systems. The number of classified sources of interest along with their general headings can be reported wirelessly to a base station every few hours via low-power wireless transceivers and Global System for Mobile (GSM) communication modems added using built-in expansion slots.

Although the EMS system can be used for any acoustic-based surveillance, situation awareness, and monitoring application, the specific application considered here is the detection and classification of man-made airborne sources (e.g., commercial and military aircraft in national parks). The goal is to identify areas of heavy noise pollution in parks and study their effects on the parks' ecosystems. Currently, detection and classification tasks are done manually by human expert operators on the previously collected data using monitoring stations deployed in some park locations. This process is very labor intensive as it requires technicians to sort through hundreds of hours of acoustic data to manually label and identify events of interest. As a result, it is impractical and perhaps impossible to deploy a large number of such stations for finer spatial and temporal resolutions monitoring and for an extended period of time (e.g., several weeks or months) without requiring to store and manually postprocess an overwhelmingly large volume of data. The EMS offers an autonomous and fully decentralized solution for near real-time soundscape characterization with minimal human involvement. This, coupled with the low-cost, low-power, and small form factor features of the EMS system will allow for large-scale deployment of these devices in national parks as well as other potential sites.

This paper is organized as follows. Section II provides a detailed description of the designed EMS system and its overall architecture, components, and many unique capabilities for soundscape characterization in national parks. Section III gives an overview of the source detection, classification, and tracking algorithms employed in this paper. These include the sparse coefficient state tracking (SCST) method [6] for simultaneous acoustic transient detection and classification, and the wideband AoA estimation algorithm [7] for source tracking and flight path determination. The details of the implementation of these algorithms on the EMS system are discussed in Section IV. The field test results of the EMS system are presented in Section V. Finally, conclusions and the future work ideas are given in Section VI.

II. EMS Hardware Architecture, Components, and Capabilities

The EMS is a new, custom-designed, multichannel, acoustic monitoring system that provides the ability to continuously monitor the audible environment and at the same time provide auxiliary data from the onboard sensor suite including temperature, pressure, light, acceleration, wind speed/direction, as well as digital compass and GPS modules. Fig. 1 shows the main components and functionalities of the EMS described next.

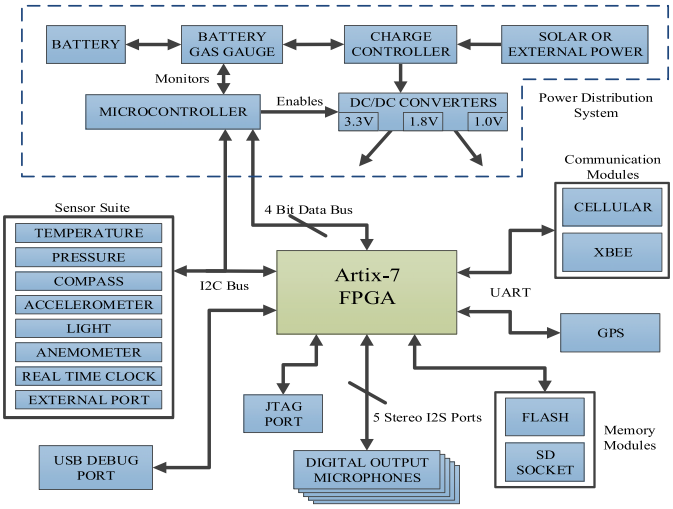

Fig. 1. EMS components and connections.

A. FPGA Processing Core

The core of the EMS is an Artix 7 FPGA that implements sensor-level one-third (1/3) octave filter bank and source detection, classification, and tracking algorithms, as well as the glue logic for the data collected by the sensors. It also implements all the drivers for the various sensors and microphones attached to the board allowing for easier modification of the design when some components need to be replaced. Since the Artix 7 used in this design has only 608 kB of RAM, an additional 32-MB flash memory chip was added to store both the bitstream used to configure the FPGA at powerup and also the various parameters that are used by the source detection and classification algorithm. In addition to the flash memory, the EMS system also has a micro SD socket that allows the Artix 7 FPGA to communicate with a standard SD card using the SPI protocol. This card can be used as backup storage for acoustic event reports in case wireless connectivity is not available to send these reports to a park station. The SD card can also store system configuration parameters that can be used to change system operations based on the application. The Artix 7 can support up to a 100-MHz sampling frequency. The system supports forming a sparse wireless sensor network of several EMS nodes via two high-performance communication links that are described in Section II-B. This allows the system to send daily summaries of the observed sources or alerts wirelessly to park headquarters.

B. Communication Modules

As shown in Fig. 1, the EMS system is designed with two expansion slots for wireless communications: one for a cellular

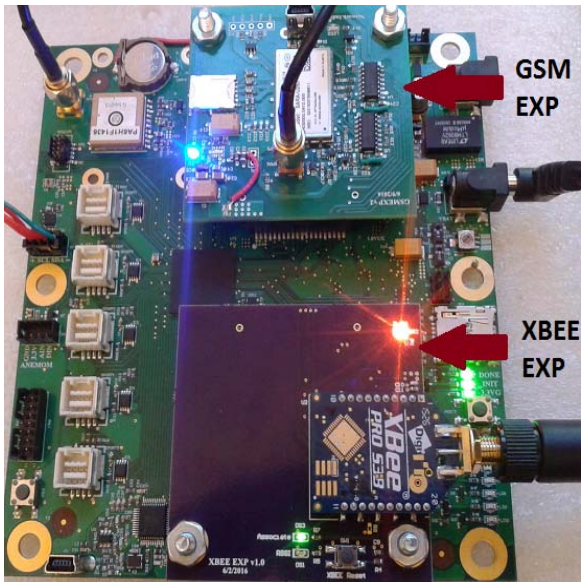

Fig. 2. EMS with GSM and XBee expansions boards.

module and the other for an XBee radio transceiver. This flexibility allows the deployed systems to use either the GSM module or the XBee radio and transmit data to a park station depending on the availability of cellular service in the deployment location and the distance between the system deployment site and the park station. The XBee module (Digi International XBee Pro S3B) is mainly used by the slave modules to communicate with the anchor (master) module in a distributed network of EMS systems. The master module then accumulates and transmits the data to the base station. The XBee must be configured in the application programming interface (API) mode to communicate the data in a structured manner and also allows communication among multiple nodes sharing information about the detected acoustic events. In addition, using the API mode, one can determine the signal strength of each received packet, dynamically estimate the link strength, gather information related to packet loss, and send firmware updates from a remote location or configure the systems in a local network, if necessary. Moreover, this communication scheme allows slave nodes to be deployed in areas with poor cellular connectivity so long as they can communicate to adjacent nodes using the XBee radios and route their data to a master node that has a good cellular connectivity. The EMS Xilinx Artix 7(XC7100T)-based main board populated with the GSM and XBee expansion prototypes are shown in Fig. 2.

Upon reception of packets sent by slave modules, the master module has to update the data in the online server. The master module has both XBee and GSM functionalities. The GSM module used in this paper is UBLOX-SARAU260 that is first configured in general packet radio service (GPRS) mode. Then, hypertext transfer protocol (HTTP) requests are sent to the server to enable the server to receive the packets sent by the EMS system. GPRS supports Transmission Control Protocol and the Internet Protocol of operation which automatically accounts for the pauses during handover and packet losses. In particular, the system uses HTTP POST messages at the application layer to transfer data between the base station and the web server. Using the POST method, the EMS system creates and sends requests to a web server to store reports contained in the body of the POST message. The server stores all report data in a structured query language database. Queries can then be made against the database for further data analysis and visualization.

C. Acoustic Channels

The EMS system has five digital stereo audio channels each of which is able to support up to two digital microphones. These microphones can be configured in an array geometry for possible AoA estimation and tracking of moving sources. The microphone used here is an InvenSense ICS-43432 that outputs 24-bit digital data sampled at 48 kHz. This microphone was selected because of its large dynamic range and low cost. The FPGA provides a clock signal along with a few other control signals to the microphone and reads in the data 1 bit at a time. These data streams are buffered until a full 24-bit sample is shifted in.

A five-microphone wagon-wheel planar array with a radius of $r = 0.098$ m was configured, where four microphones were spaced evenly around the perimeter of the array and one microphone (reference) was in the center. A custom adapter PCB was designed for the microphones, which included connectors, buffers, and terminations for the signal lines. These microphone adapters were used in the construction of the array. Since the drivers for the microphones are implemented on the FPGA, most types of digital output microphones can be used with only minor modifications to the HDL code. This allows flexibility in selecting microphones with characteristics suited for specific applications. The array shape was decided based on its ease of manufacturing and source characteristics. This planar wagon-wheel array allows for accurate azimuth angle estimation of the received acoustic signals although the estimate of the elevation angle is only approximate. This is adequate for our purpose as we are only concerned with determining the general headings of the airborne sources. If a more accurate estimate of elevation angle is needed, the design can be improved by elevating some of the microphones and forming a volume array.

D. Environmental Sensors

In addition to streaming acoustic data, the EMS system can also collect general-purpose data about the environment in which it is deployed via a suite of onboard sensors shown in Fig. 1. The compass can provide a precise orientation of the system in order to produce accurate AoA estimates. The accelerometer will be used to determine the orientation of the system and identify if it has been knocked over by wind or an animal. The light sensor can be used for power management by powering down some nonessential systems when there is not much sunlight available to operate the system and charge the battery. The temperature, pressure, and anemometer will provide useful information about the deployed environment which may be reported along with other information to a park station. In addition, the *in situ* measurements of the temperature and wind velocity profiles may be used to estimate

the speed of sound within the deployment area which can in turn be used by the beamforming algorithm to obtain more accurate AoA estimates.

E. Power Systems

The EMS is intended to be deployed in remote locations, and therefore, it is designed to be powered by batteries. However, for long-term deployments, the system is also able to accommodate a renewable energy source like solar power. It includes charge controller circuitry so that excess power from the solar panel can be used to recharge the battery. With the solar panel setup, if there is enough insolation during the day, the EMS system can operate continuously for a long period of time by using solar power during the day and relying on battery power at night. To achieve a minimum current draw and proper input/output states of the FPGA during ON-power, an Atmel ATtiny88 microcontroller is used to enable the system dc/dc converters in a particular sequence according to the switching characteristics specified for this FPGA by the manufacturer. This controller can also monitor the amount of battery charge remaining via an onboard battery power gauge. This information can be used to put the system in sleep mode if the remaining battery charge gets critically low during minimal insolation periods. The microcontroller and FPGA are linked via a 4-bit communication bus so that data can be easily shared between the two.

F. Localization and Synchronization

A GPS module and a real-time clock module are also included for time synchronization and node localization. The real-time clock is periodically synchronized to the time reported by the GPS in order to prevent clock drift. This real-time clock in turn is used to timestamp acoustic events that are detected and reported to the park station. Precise time synchronization is necessary not only for AoA estimation but also for fusing data between multiple EMS systems deployed in a park. If a GPS signal is not available in some deployment locations but time synchronization is still necessary, then a network-based synchronization protocol can be implemented. The GPS will also be able to provide accurate location information (to within 3 m of the receiver) for each of the deployed nodes. This information will be integrated into the reports sent by each EMS system allowing park technicians to easily identify where each acoustic event was observed and to visualize the network of deployed EMS nodes on a map.

G. Health Monitoring

The EMS is also capable of node health monitoring. Each node can send out periodic reports containing subsystem operational status. Nodes can report failures in the acoustic hardware, battery charging, and sensor subsystems. A node that fails to send out the periodic health status update indicates a problem with the communication modules, a power failure, or a major hardware fault. When the nodes are distributed forming a mesh network, the system is still capable of sending reports even if a node turns OFF due to the minimal power availability or hardware failure. In a mesh topology, if a node is no longer able to operate, the remaining nodes can still communicate with each other if there are redundant communication paths. EMS health monitoring capabilities allow technicians to rapidly evaluate the health of the entire network and pinpoint nodes that require service.

All the above-mentioned components work together to provide timely, in-depth reports of the environments in which the systems are deployed to the park stations. These reports can be used to create soundscapes of these areas without having to manually analyze acoustic data.

III. Overview of Source Detection, Classification, and Tracking Algorithms

As mentioned earlier, the key benefit of the EMS system is its ability to perform near real-time and fully decentralized detection and classification of airborne acoustic sources of interest. Once a source is detected and classified, the EMS can run a sensor-level wideband AoA estimation using the captured acoustic data of all the microphones in an array. The general headings of the sources generated using the AoA estimates together with their class labels and timestamps are logged into a file that is periodically transmitted to a park station. Fig. 3 illustrates all these processing steps implemented on the EMS. In this section, we provide an overview of the SCST-based source detection and classification algorithm and then briefly discuss the procedure for source AoA estimation and tracking.

A. SCST-Based Source Detection and Classification Method

Many algorithms have been proposed for detection and classification of acoustic transient sources. A thorough review of these methods is provided in [6]. Here, we adopted the SCST [6] to perform fully decentralized and simultaneous detection and classification of transient acoustic events. This is done by continuously monitoring the streaming multivariate (e.g., 1/3 octave) acoustic data and detecting the start and the end of an event while at the same time identifying its source type. In the following, we briefly describe different processes in the SCST method.

Separating the transient events of interest from the ambient noise and other noninteresting data requires two phases: 1) signal detection to locate the presence of a transient signal of an unknown source under the assumption that none were present and 2) quiescent detection to find the endpoint of the transient signal by searching for observations where the particularly dominant source is no longer present under the assumption that there was one present. This approach assumes that each unique transient has a finite extent and is separated by a distinct, albeit variable length, quiescent period. If the quiescent period does not exist, the overlapping sources of interest will be merged into a single event and the less dominant source will be missed.

When the data have been in a quiescent period since time $\hat{k}_0$, signal detection can be accomplished for each new feature vector $\mathbf{z}_k$ using the following multiple hypothesis tests:

$$\begin{aligned} \mathcal{H}_0 &: \ \mathbf{z}_k = \mathbf{w}_k, \quad \hat{k}_0 \le k \le n \\ \mathcal{H}_1^{(p)} &: \ \mathbf{z}_k = \begin{cases} \mathbf{w}_k, & \hat{k}_0 \le k \le k_1 \\ \mathbf{s}_k^{(p)} + \mathbf{w}_k, & k_1 \le k \le n \end{cases} \end{aligned} \tag{1}$$

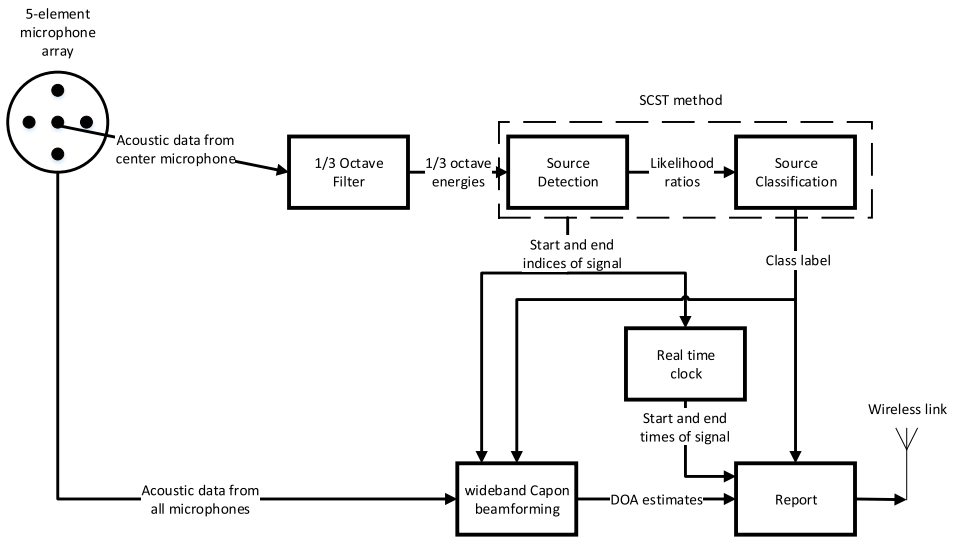

Fig. 3. Overview of acoustic signal processing on EMS board.

where $\mathbf{s}^{(p)}$ is a source vector of an unknown class $p \in [1, P]$ with P being the total number of considered source classes, and $\mathbf{w}_k$ is an ambient background noise vector. The null hypothesis $\mathcal{H}_0$ and alternative hypothesis $\mathcal{H}_1^{(p)}$ represent the absence and the presence of any prominent transient source labeled p, respectively. The times k_0 and k_1 denote the onset times for the next unknown quiescent and source periods, respectively, while $\hat{k}_0$ and $\hat{k}_1$ denote the estimates of the most recently observed quiescent and detection periods, respectively.

Given that the acoustic transient event has a finite extent, quiescent detection is needed to identify the endpoint of the transient source or onset of the next quiescent period. When a source signal has been present since time $\hat{k}_1$, the following test is used to perform quiescent detection:

$$\begin{aligned} \mathcal{H}_1^{(p)}:\ \mathbf{z}_k &= \mathbf{s}_k^{(p)} + \mathbf{w}_k, \quad \hat{k}_1 \le k \le n \\ \mathcal{H}_0:\ \mathbf{z}_k &= \begin{cases} \mathbf{s}_k^{(p)} + \mathbf{w}_k, & \hat{k}_1 \le k \le k_0 \\ \mathbf{w}_k, & k_0 \le k \le n \end{cases} \end{aligned} \tag{2}$$

where $\mathbf{s}_k^{(p)}$s cease to be extant at the unknown time k_0 under the null hypothesis $\mathcal{H}_0$.

To implement the hypothesis test in (1) on streaming quantized data vectors, $\mathbf{z}_n$, and provide a test statistic for signal detection and evaluating the relative likelihoods of $\mathcal{H}_1^{(p)}$, we adopt a cumulative test statistic given as

$$B_p(n) = \max\{0,\ B_p(n-1) + b_p(n)\}, \quad n = \hat{k}_0, \hat{k}_0 + 1, \ldots \tag{3}$$

which is initialized as $B_p(\hat{k}_0 - 1) = 0,\ \forall p$. This statistic is updated using

$$b_p(n) = \begin{cases} \ln\left(\dfrac{f_{\lambda_p}(\mathbf{z}_n|\mathbf{z}_{n-1})}{f_{\lambda_0}(\mathbf{z}_n)}\right), & B_p(n-1) > 0 \\ \ln\left(\dfrac{f_{\lambda_p}(\mathbf{z}_n)}{f_{\lambda_0}(\mathbf{z}_n)}\right), & B_p(n-1) = 0 \end{cases} \tag{4}$$

where $f_\lambda(.)$ is a probability distribution modeled by the parameter set $\lambda \in \{\lambda_0, \lambda_p\}$ containing noise λ_0 and source λ_p parameters. Since λ_p is unknown, our test determines which unknown source parameter set is the most likely one

$$\max_p\ B_p(n) \ge \eta \tag{5}$$

where the cumulative value of $B_p(n)$ simply must exceed the detection threshold η for any source label p.

Once a transient signal is declared, the quiescent detection uses the following cumulative test statistic:

$$T_p(n) = \max\{0,\ T_p(n-1) + t_p(n)\}, \quad n = \hat{k}_1, \hat{k}_1 + 1, \ldots \tag{6}$$

initialized as $T_p(\hat{k}_1 - 1) = 0,\ \forall p$ to determine the endpoint. This test statistic is updated using

$$t_p(n) = \ln\left(\frac{f_{\lambda_0}(\mathbf{z}_n)}{f_{\lambda_p}(\mathbf{z}_n|\mathbf{z}_{n-1})}\right). \tag{7}$$

Unlike $b_p(n)$, the value of $t_p(n)$ does not depend on the value of the corresponding cumulative test statistic, $T_p(n-1)$, at time $n-1$ since conditional distributions are always used under $\mathcal{H}_1^{(p)}$ in the quiescent detection phase. The absence of any source is declared at time n whenever

$$T_{p^*}(n) \ge \gamma \tag{8}$$

where

$$p^* = \arg\max_p\ B_p(n) \tag{9}$$

and γ is the threshold for quiescent detection and p^* represents the determined class label of the detected source at time $\hat{k}_0 - 1$. Thus, in the SCST algorithm, successful signal and quiescent detection are the prerequisites for making a correct transient classification. That is, once the end of a source is declared, the source type that has the highest test statistic $B_p(n)$ at that time represents the class of that source. The process then reverts back to look for a new source of an

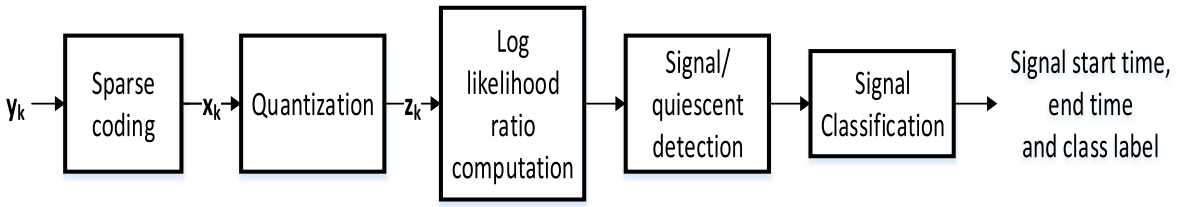

Fig. 4. SCST block diagram.

unknown type according to (1). This phase switching process continues indefinitely, logging the detected source each time a new quiescent period starts.

To implement the SCST tests in (5) and (8), one needs to compute the probability density functions (pdfs) $f_{\lambda_0}(\mathbf{z}_n)$, $f_{\lambda_p}(\mathbf{z}_n)$, and $f_{\lambda_p}(\mathbf{z}_n|\mathbf{z}_{n-1})$ to update $B_p(n)$ and $T_p(n)$, $\forall p \in [1, P]$ for every new observation vector $\mathbf{z}_n$. This requirement is generally intractable without assuming statistical independence of observations $\mathbf{z}_n$s under each hypothesis $\mathcal{H}_1^{(p)}$. The SCST algorithm uses a Bayesian network [9] that efficiently computes these pdfs without requiring independence of the observations. This is done by decomposing each pdf into a product of constituent conditional probabilities given other dependent states, which can be computed using the lookup tables established during the training phase. For more detailed information on the Bayesian network training and implementation, the reader is referred to [6].

Fig. 4 shows the block diagram of the SCST implementation for this problem. First, the 1/3 octave spectral features extracted from the acoustic data in 1-s interval are sparsely coded using a composite dictionary matrix formed from the source and interference dictionary matrices. To filter the contribution of the superimposed interference and noise, the sparse coefficients corresponding to the interference dictionary atoms are discarded. The retained coefficients associated with the sources are then quantized using a discriminative quantizer [6]. The quantization is done to reduce the number of possible states and hence the overall computations. The sparse coded and quantized *feature vector* is applied to a trained Bayesian network, which generates the conditional probabilities of the feature vector given the previous observations under each source and interference models. The log-likelihood ratios for each source are then computed and added to their respective running sums that are subsequently compared against preselected thresholds in (5) and (8) to determine the start and end times of the transient sources. Once the end of a source signal is decided, the classification is done based on the same cumulative sums.

B. Geometric Wideband Capon Method

In this paper, we adopted the geometric averaged wideband Capon algorithm mainly owing to its reduced computational requirements [7]. This algorithm computes the AoA of the source wavefront using the recorded data from all the microphones on the array in every 1-s snapshot. Each 1-s snapshot from each microphone is partitioned into K nonoverlapping blocks of length 1024 samples, and fast Fourier transform (FFT) is applied to each zero-padded block. The transformed vector for the kth block at narrowband frequency component ω_j, $j \in [1, J]$ is denoted by $\mathbf{x}_k(\omega_j)$, $k = 1, 2, \ldots, K$. These narrowband components are used to compute the sample *spatial covariance matrix*

$$\mathbf{R}_{\mathbf{xx}}(\omega_j) = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_k(\omega_j)\mathbf{x}_k^H(\omega_j). \tag{10}$$

The spatial covariance matrices for all the frequency bins ω_j, $j \in [1, J]$ of the classified source are used to generate the geometrically averaged wideband Capon power spectrum [7]

$$\mathbf{Q}_G(\theta) = \prod_{j=1}^{J}\frac{1}{\mathbf{v}^H(\omega_j,\theta)\mathbf{R}_{\mathbf{xx}}^{-1}(\omega_j)\mathbf{v}(\omega_j,\theta)} \tag{11}$$

where $\mathbf{v}(\omega_j, \theta)$ is the array steering vector and θ is the azimuth angle relative to the microphone array. For our five-microphone wagon-wheel array, $\mathbf{v}(\omega_j, \theta) = [e^{-j(\omega_j r/c)\sin\theta}, e^{-j(\omega_j r/c)\cos\theta}, 1, e^{j(\omega_j r/c)\sin\theta}, e^{j(\omega_j r/c)\cos\theta}]$, where r is the radius of the array and $c = 344\ m/s$ is the speed of sound in air. This steering vector assumes that the microphone inputs are ordered as 1: East, 2: South, 3: Center, 4: West, and 5: North.

This aggregated power spectrum is then searched over the azimuth angle θ and the angles that maximize this function are determined to be the AoA angles of the detected sources for that snapshot. This procedure is repeated for every 1-s snapshot while the source is in the audible range in order to produce successive AoA estimates that can be interpolated to form the flight path of the source.

IV. Source Detection, Classification, and Tracking on the EMS

A. SCST Implementation

The SCST algorithm was implemented on the Artix 7 FPGA as a mix of custom logic and software running on a soft-core processor. The bulk of the computations in the SCST algorithm comes from performing sparse coding for each 1/3 octave spectral vector while other computationally costly steps are done offline during the training phase. Once the sparse coding and quantization of the sparse coefficients are done, lookup tables of the trained Bayesian network are used to compute the relevant test statistics for detection and classification processes. Fig. 5 depicts several steps in the SCST implementation, which are discussed in the following.

1) 1/3 Octave Filter Bank: Once the 24-bit acoustic data stream is acquired from the digital microphones (step 1 in Fig. 5), the 1/3 octave filter bank is implemented on the FPGA to extract 33 subband features (from 100 Hz to 1 kHz)

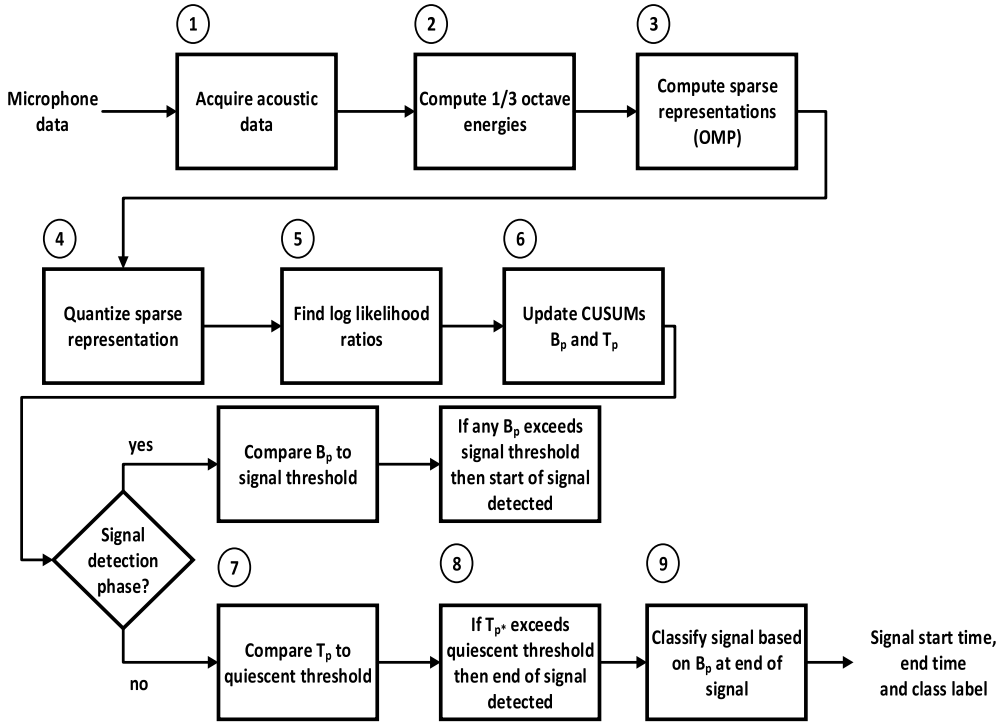

Fig. 5. High-level overview of SCST implementation.

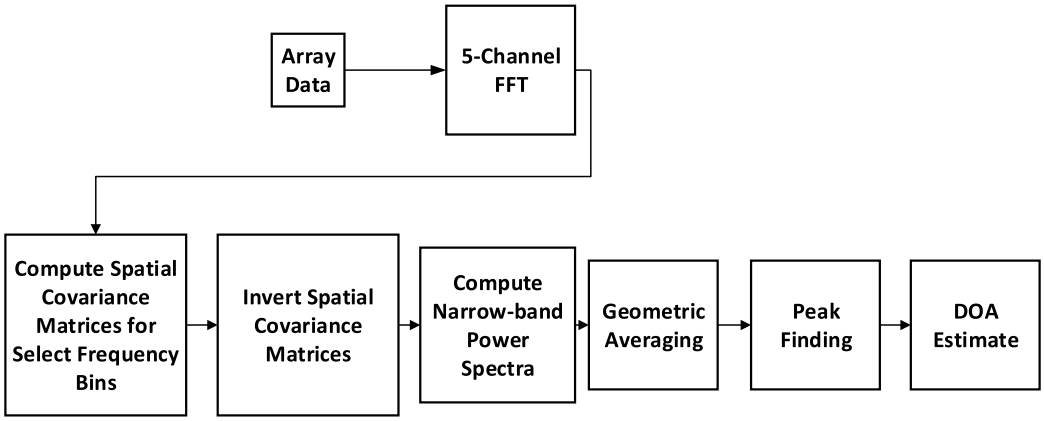

Fig. 6. Overview of wideband capon beamforming implementation.

for every 1-s snapshot. This filter bank implementation is based on the iterative method in [10], which consists of four different digital filters. The first three filters are eighth-order infinite impulse response (IIR) Butterworth digital filters implemented using four second-order sections with center frequencies of 0.488π, 0.625π, and 0.781π. These filters decompose the incoming signal into three 1/3 octave components that meet IEC 61260 Ed. 1.0 [11] requirements for a Class 0 device. The last filter is an eighth-order IIR low-pass filter (also implemented using four second-order sections) with a cutoff frequency of 0.438π. This filter is applied to the signal after the first three filters and then the signal is downsampled by a factor of two. This allows the same first three filters to be used again on the downsampled signal in order to separate out the 1/3 octave content corresponding to the next octave. The downsampling is done in real-time with a counter specifying which outputs of the low-pass filter to be ignored. This process is repeated 11 times in order to create 33 timeseries, each corresponding to a different 1/3 octave frequency subband. The total signal energy over every 1-s interval is then computed for each of the 33 subband outputs and the results are stored in registers. An interrupt signal is sent to the Microblaze processor. When the processor receives the interrupt signal, it reads the registers and retrieves these energy values. The processor then starts sparse coding, signal detection, and classification (steps 3–9 in Fig. 5) and waits for the next set of energy values to be ready.

2) Sparse Coding: The sparse coding in step 3 is done using a fast orthogonal matching pursuit (OMP) algorithm [12] on the Microblaze soft-core processor instantiated in the Artix 7 FPGA fabric. This algorithm was chosen as it avoids costly matrix inversion inherent in the original OMP method. The dictionary matrix utilized in the fast OMP is generated using the K-SVD algorithm [13] during the training phase. This dictionary matrix is stored in RAM on the Microblaze as

a 2-D double precision array. All the other computations are done using the floating point processing unit on the Microblaze and a double precision multiplier implemented in custom logic. The computation of the inner products as well as the iterative updating of the residual, least squares filter, and other variables used by the fast OMP algorithm are all computed using for-loops.

3) Quantization: After using fast OMP on each 1/3 octave data vector, $\mathbf{x}_n$, the coefficients associated with interference dictionary atoms are first removed (filtered) and then the *coefficient state vector*, $\mathbf{z}_n$, is generated by elementwise quantization of the remaining coefficients in this vector using predetermined quantization levels as shown in Fig. 4. The quantization threshold is selected during the training to force coefficients likely attributed to ambient background noise to zero state. The transition levels are also determined during the training to maximize interclass discrimination among different classes [6]. The number of quantization levels L is chosen such that the quantized states evolve adequately for interclass discrimination yet L is also kept relatively small since the quantizer resolution drives the state distribution model size. Our experimental results indicated that a quantizer with $L = 4$ levels was a suitable choice for this problem.

4) Detection and Classification: After the quantization, the detection and classification processes are carried out by first finding the log-likelihood ratio of the coefficient state vector, $\mathbf{z}_n$, having been produced by a certain source model versus the noise alone case using the lookup tables. It does this by looking up the log-likelihood of each of the elements in $\mathbf{z}_k$ being from a certain source model and adding these values together using the process described in Section III-A. If any running total $B_p(n)$ for a source type exceeds a preset threshold η, then a source is detected. For quiescent detection, the log-likelihood ratios $t_p(n)$s are computed and accumulated for each source [(see (6))] similar to $b_p(n)$. When the cumulative ratio $T_{p^*}(n)$ of the source type p^* that has the highest cumulative source detection ratio $B_{p^*}(n)$ exceeds a preset threshold γ, then the end of the source is detected and the source is classified as being of class p^* as in (9). Afterward, all the cumulative ratios (B_p and T_p) are reset to zero and the algorithm starts looking for the start of a new source signal.

The log-likelihoods of each entry in the sparse-coded quantized vector $\mathbf{z}_n$ are stored in a lookup table in flash memory instead of RAM because it takes a lot of space (over a megabyte) and only a few entries from this table need to be read every second. However, the number of dependencies of each entry under each source model is stored in an array in RAM so that the algorithm can quickly find the correct log-likelihood value for each entry.

B. Wideband Capon Implementation

The wideband Capon beamforming implementation consists of several steps that are shown in Fig. 6. These are described in the following.

1) Data Preparation: When a source is detected and classified, the detected segments of data acquired by all the microphones in the array are arranged into a multivariate timeseries which is then applied to a multichannel FFT core. It then outputs the results of this 1024-point FFT for each sequence one frequency bin at a time to a first-in first-out (FIFO) type buffer starting with the lowest frequency and ending with the highest.

2) Computing Spatial Covariance Matrices: Whenever a set of FFT outputs is available in the buffer, it is read by the spatial covariance matrix computation state machine implemented in custom logic on the Artix 7 FPGA. Since these complex-valued data in the FIFO buffer are ordered by the frequency bin for which they were generated, this state machine keeps reading in these values from the buffer until it reads in a complete set that is associated with each frequency of interest of the classified source. It orders these values into a vector that is then multiplied by its transpose to form a rank-one matrix for each frequency bin. This rank-one matrix is then added to the matrix that was already stored in the RAM on the EMS board for this frequency bin. Once a specific number (here $K = 32$) of rank-one matrices for each frequency bin have been summed, the entries of all the matrices in RAM are normalized by K using a divider core as in (10). The resulting sample covariance matrices for all frequency bins are output to another FIFO buffer 1 entry at a time.

3) Matrix Inversion: When data are available in the FIFO buffer, they are read in by the matrix inversion state machine, also implemented as custom logic in the FPGA, until an entire covariance matrix is read in. It then converts each entry in the matrix into double precision using a Xilinx core in order to avoid possible overflow/underflow issues during the inversion process. It then computes the inverse spatial covariance matrix using a modified Gram–Schmidt method outlined in [14]. The entire inversion process is repeated for each matrix in the spatial covariance buffer and the resulting inverse covariance matrices are stored in RAM in the same order in which they were processed, i.e., lowest to the highest frequency. Since a full set of covariance matrices (one for each frequency bin) is stored in the spatial covariance FIFO buffer every second, this state machine inverts all of these covariance matrices and stores them in RAM every second overwriting the covariance matrices from the previous snapshot.

4) Wideband Beamforming: Once the inverse spatial covariance matrices are stored in RAM, a signal is sent to the Microblaze processor to initiate the wideband beamforming process at which point it premultiplies each inverse matrix by the Hermitian of the array steering vector at a particular azimuth incidence angle and then postmultiplies the result by the steering vector itself. This process is repeated for every angle over which the spectrum is to be searched resulting in an inverse of the narrowband Capon power spectrum for each frequency bin of interest. These inverted power spectra are then multiplied together to produce a single geometrically averaged wideband power spectrum according to (11). This inverted power spectrum is then searched for its minimum value and the angle at which this minimum value occurs is declared as the AoA of the source of interest. This procedure is repeated at every snapshot and the resulting sequence of AoA estimates is then logged so that it may later be used to generate the path of the moving source.

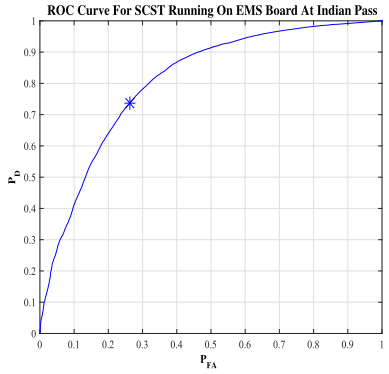

Fig. 7. ROC curve of SCST-based detector.

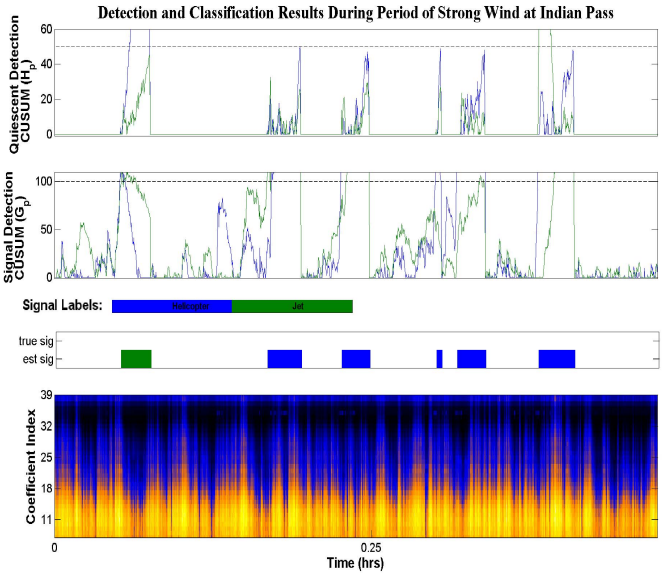

Fig. 8. Detection and classification results during strong wind.

V. Test Results of a Field Deployed System

A. Test Results of SCST Implementation

An EMS system was recently deployed at the Lake Mead National Recreation Area. The acoustic data captured by only one microphone were used in this study. The system was first trained based upon annotated and labeled 1/3 octave data previously collected over a 48-h period at the same location. The training involved building the dictionary matrices using the K-SVD algorithm [13] for different source types and interference present at this particular site and computing the log-likelihood values for each Bayesian network source model. These dictionary matrices and log-likelihood values were then stored in the memory of the EMS.

The system was specifically trained to detect and classify helicopters and jets as these were the dominant sources of interest present at this particular site. The helicopter signals were much more prevalent when compared to jets as this location is a high-traffic area for air tours. Moreover, the helicopter signals were of much higher intensity than the jet signals and thus easier to identify even in windy conditions. This is likely because the helicopters generally fly at much lower altitudes. The only prevalent interference happened to be the wind as there were no other natural sounds present during the time that the system was deployed. The wind intensity, however, varied significantly throughout the test and even with a windscreen, the wind would at times be strong enough to saturate the microphone.

TABLE I
Confusion Matrix of SCST-Based Classifier

Truth \ Decision	Helicopter	Jet
None	$\frac{73}{208}$	$\frac{135}{208}$
Helicopter	$\frac{237}{271}$	$\frac{34}{271}$
Jet	$\frac{25}{73}$	$\frac{48}{73}$

The trained system was then tested for another 48-h period when deployed at the same location to perform sensor-level source detection and classification. The signal and quiescent detection thresholds were chosen experimentally to be $\eta = 100$ and $\gamma = 50$, respectively. The receiver operating characteristic (ROC) curve for the SCST-based detector is shown in Fig. 7. It must be noted that this ROC curve was generated based on each individual observation (as opposed to the entire transient event), i.e., the probability of detection for each point on the curve was calculated using the number of 1/3 octave vectors that were correctly determined by the SCST to have contained a source based on the ground truth data. Likewise, a false alarm probability was calculated using the number of 1/3 octave vectors that were incorrectly determined by the SCST to have contained a source based on the ground truth data. The reason the ROC curve was generated this way is that since the sources can span a various number of observations, as the detection threshold is decreased the number of false detections does not necessarily change monotonically. As new false detections appear in the data, old ones might combine together due to the observations between them exceeding the detection threshold. Therefore, it would be difficult to accurately calculate a meaningful percent false alarm rate based on the number of falsely detected sources. The knee point of the ROC is at which $P_{FA} + P_D = 1$ exhibits $P_D = 74\%$ and $P_{FA} = 26\%$. Most of the false alarms occurred during periods of intense wind when the microphone was saturated and the 1/3 octave filter reported high energy content in all frequency bins. This can be seen in a small 1/2-h segment of the collected data in Fig. 8 where the top and middle plots show the quiescent and signal detection statistics as a function of time while the bottom figure gives the plot of the 1/3 octave spectral features.

Table I, on the other hand, gives the confusion matrix of the SCST-based classifier. As can be seen, the majority of the 271 helicopter instances that were detected during the deployment were correctly classified. However, the system had more trouble correctly classifying the jet sources. This is primarily due to the lack of adequate training samples for the jet type when compared to the number of helicopter samples and more importantly the low signal-to-interference ratio for jet sources particularly when the wind was strong. It also falsely detected and classified 135 instances of jets when there were no sources present in the data. This can be seen in Fig. 8 which shows the false classifications of

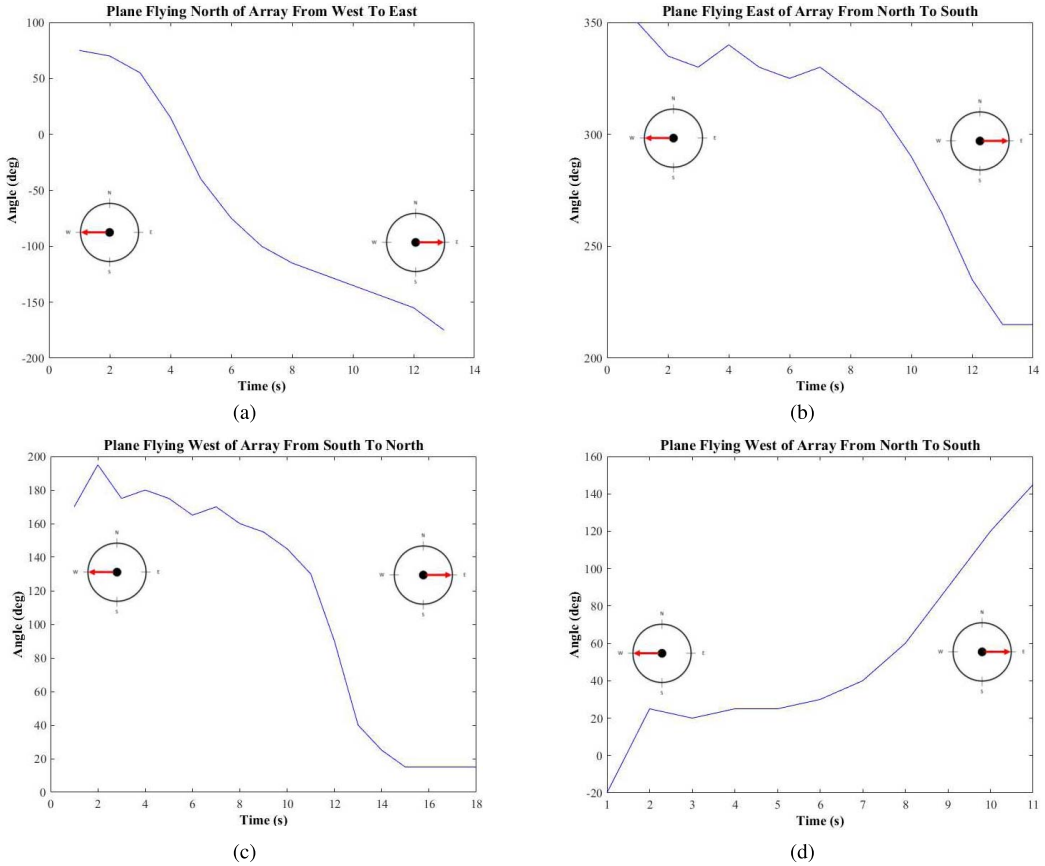

Fig. 9. AoA estimate tracks of a model plane for four different headings.

jet and helicopter sources in the presence of strong wind. This is due to the fact that while the energy in the lower frequency bins are likely attributed to the wind and removed by the sparse coding, the energy present in the mid and high-frequency bins would most certainly look much more like helicopter signal model rather than the noise model where energy is distributed about evenly between all frequency bins. It was also noted even in the absence of strong wind, some of the jet sources were quite a bit weaker than the helicopter sources; hence, they were much harder to distinguish from the interference.

B. Test Results of Wideband Beamforming Implementation

A controlled full system test was conducted using the wagon-wheel microphone array described in Section II for AoA tracking of a small model airplane to determine the general headings relative to the array. The AoA estimates produced by the wideband Capon beamforming implemented on the EMS were recorded and used to create estimates of the flight paths of the plane, which were then compared against the actual headings at which the plane was flown. This was repeated several times with different headings relative to the array. Fig. 9(a)–(d) shows the azimuth angle tracks formed by the wideband Capon beamforming implemented for several different headings. The compasses on these figures show the actual starting and ending locations (relative to the array) of the plane for each track. These tracks show that the AoA estimates were able to capture the general direction the plane was flying in even though a few estimates were erroneous. These erroneous estimates were likely caused by interference, e.g., strong wind, especially at long ranges from the array when the signal-to-noise ratio (SNR) was low. Even with these errors, the estimated trajectories are acceptable for our application as the main goal is to know in which direction the detected sources of interest are flying.

VI. Conclusion

This paper introduced a new FPGA-based platform, referred to as the EMS system, for fully decentralized detection, classification, and tracking of man-made acoustic sources. The system is custom designed and prototyped around an Artix 7 FPGA equipped with an array of microphones, an environmental sensor suite, two different communication modules, power management options, GPS-based synchronization, and memories. Source detection and classification on the EMS implement a bank of filters for the 1/3 octave subband feature extraction, sparse coding, and quantization to filter out the interference and noise and reduce the number of states. The SCST algorithm implemented on the EMS then performs simultaneous source detection and classification based upon the resultant features. Once a source is detected and classified, a wideband beamforming algorithm, also implemented on the EMS, enables one to produce estimates of the flight path of the airborne sources. A 48-h field test was conducted at the Lake Mead National Recreation Area using one EMS to examine the detection and classification performance of the system where the main sources of interest were jets and helicopters. The system provided samplewise

$P_D = 74\%$ and $P_{FA} = 26\%$ detection performance at the knee-point of ROC curve that is found to be comparable to human expert's performance. As for source classification, the system provided adequate correct classification rates though misclassification and false alarms did occur more frequently for jet sources. This may be attributed to several reasons: 1) the imbalance in the number of samples for helicopters and jets at this particular site; 2) the relatively low SNR of jet signatures when compared with those of the helicopters; and 3) the presence of strong wind which not only masks the weak sources but also caused microphone saturation. Although the EMS system was used here mainly for noise monitoring in national parks, it can also be applied to numerous other important applications such as detection and localization of gunshots, detection and tracking of airborne and ground vehicles, traffic monitoring, and in security systems.

ACKNOWLEDGMENT

The authors would like to thank Dr. K. Fristrup from National Park Service in Fort Collins, CO, USA, for all his invaluable help and suggestions throughout this research project. This work could not have been completed without his contributions.

REFERENCES

[1] O. Postolache, P. Girao, and M. Pereira, "Underwater acoustic source localization based on passive sonar and intelligent processing," in *Proc. IEEE Instrum. Meas. Technol. Conf. (IMTC)*, May 2007, pp. 1–4.

[2] F. Pianegiani, M. Hu, A. Boni, and D. Petri, "Energy-efficient signal classification in ad hoc wireless sensor networks," *IEEE Trans. Instrum. Meas.*, vol. 57, no. 1, pp. 190–196, Jan. 2008.

[3] J. Sallai, W. Hedgecock, P. Volgyesi, A. Nadas, G. Balogh, and A. Ledeczi, "Weapon classification and shooter localization using distributed multichannel acoustic sensors," *J. Syst. Archit.*, vol. 57, no. 10, pp. 869–885, 2011.

[4] P. E. William and M. W. Hoffman, "Classification of military ground vehicles using time domain harmonics' amplitudes," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 11, pp. 3720–3731, Nov. 2011.

[5] S. Akhtar, M. Elshafei-Abmed, and M. S. Ahmed, "Detection of helicopters using neural nets," *IEEE Trans. Instrum. Meas.*, vol. 50, no. 3, pp. 749–756, Jun. 2001.

[6] N. Wachowski and M. Azimi-Sadjadi, "Detection and classification of nonstationary transient signals using sparse approximations and Bayesian networks," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 22, no. 12, pp. 1750–1764, Dec. 2014.

[7] M. R. Azimi-Sadjadi, A. Pezeshki, and N. Roseveare, "Wideband DOA estimation algorithms for multiple moving sources using unattended acoustic sensors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 4, pp. 1585–1599, Oct. 2008.

[8] B. Chen and P. Willett, "Detection of hidden Markov model transient signals," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 4, pp. 1253–1268, Oct. 2000.

[9] F. V. Jensen, *Bayesian Networks and Decision Graphs*. New York, NY, USA: Springer-Verlag, 2001.

[10] A. Lozano and A. Carlosena, "DSP-based implementation of an ANSI S1.11 acoustic analyzer," *IEEE Trans. Instrum. Meas.*, vol. 52, no. 4, pp. 1213–1219, Aug. 2003.

[11] *Electroacoustics—Octave-Band and Fractional-Octave-Band Filters—Part 1: Specifications*, document IEC 61260-1:2014, 2014.

[12] M. R. Azimi-Sadjadi, J. Kopacz, and N. Klausner, "K-SVD dictionary learning using a fast OMP with applications," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 1599–1603.

[13] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[14] A. Irturk, S. MirZaei, and R. Kastner, "An efficient FPGA implementation of scalable matrix inversion core using QR decomposition," Univ. California San Diego, La Jolla, CA, USA, Tech. Rep. CS2009-0938, 2009.

Vladimir Yaremenko received the B.S. degree in electrical engineering from the Colorado School of Mines, Golden, CO, USA, in 2010, and the M.S. degree in electrical engineering from Colorado State University, Fort Collins, CO, USA, in 2017.

He is currently an Electrical Engineer with Raytheon Missile Systems, Tucson, AZ, USA. His current research interests include signal and image processing, machine learning, and real-time algorithms.

Mahmood R. Azimi-Sadjadi (LM'18) received the M.S. and Ph.D. degrees in electrical engineering from the Imperial College of Science and Technology, University of London, London, U.K., in 1978 and 1982, respectively, with a focus on digital signal/image processing.

He is currently a Full Professor with the Electrical and Computer Engineering Department, Colorado State University, Fort Collins, CO, USA, where he is also the Director of the Digital Signal/Image Laboratory. His current research interests include statistical signal and image processing, machine learning and adaptive systems, target detection, classification and tracking, sensor array processing, and distributed sensor networks.

Dr. Azimi-Sadjadi served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON NEURAL NETWORKS.

Jarrod Zacher received the B.S. degree in electrical engineering and the B.S. degree in computer science from Colorado State University, Fort Collins, CO, USA, in 2009 and 2017, respectively.

He is currently an Embedded Software Engineer with Emerson Automation Solutions, Boulder CO, USA. His current research interests include printed circuit board design, embedded systems, and firmware design.