Detection of fleeting radio signals in a congested spectrum

By David J. Sadler

Roke Manor Research Ltd., Old Salisbury Lane, Romsey, Hampshire, SO51 0ZN, UK

Abstract

A compressive sensing based technique to decompose radio spectra into background signals of little interest and foreground signals of new emissions is proposed. The algorithm is executed on-line so that data is processed immediately after it is generated in real-time, i.e. there is no significant latency whilst a large block of data is collected. The approach directly operates on wideband spectral data and is completely blind in the sense that no a priori knowledge of the foreground signals' time of appearance, centre frequency, bandwidth or modulation type is required. The mathematics of the five stages of the algorithm are presented, and the mechanisms behind the operation of the algorithm are discussed. Experimental results from processing off-air recordings are provided to demonstrate the algorithm's efficacy.

1. Introduction

Rapid detection of fleeting signals within a congested electromagnetic spectrum is a challenging problem relevant to both defence and national security. Of particular interest is the detection of previously uncharacterized signals which appear only briefly.

Detection of known signals is classically achieved by application of a matched filter to maximize Signal-to-Noise Ratio (SNR) before comparing the output level to a threshold. Such an approach is feasible if the signal characteristics are known in advance so that the appropriate matched filter can be designed. When the signal characteristics are unknown the matched filter cannot be used; instead, blind detection[1] of anomalous, short duration signals is required.

Every Signal of Interest (SOI) cannot be expected to be centred on a known frequency, nor occupy any particular bandwidth. Furthermore, SOIs may be co-channel or adjacent channel with other unknown SOIs or background interference signals. Although a blind, narrowband signal detector such as a radiometer may be scanned across a wide range of frequencies, the results will be unsatisfactory for a number of reasons: low probability of intercept due to scanning, high missed detection rate due to mismatch between the radiometer and SOI bandwidth, and an inability to distinguish between SOIs and interference. Instead, what is required is an approach that directly operates on wideband data; with the ability to separate short duration SOIs from a background of fixed frequency interference sources.

The proposed solution to the problem is inspired by signal processing methods for video streams used to separate foreground image layers, such as moving people and objects, from static background image layers. By way of comparison, our requirement is to separate the rapidly appearing foreground SOIs from a largely stationary background of interference signals and noise. Figure 1 is a diagrammatic example to explain the desired outcome. On the left hand side of the figure, three blue plots represent the received spectra at three distinct times. There are three background signals which are present at all times, but at time t_2 a fourth (highlighted) signal makes a brief appearance. This SOI is to be extracted from the spectrum. On the right

[1] Blind signal processing refers to methods where no a priori knowledge of the waveforms is presumed, i.e. the time of appearance, centre frequency, bandwidth and modulation type are considered to be unknowns.

hand side of Figure 1 the ideal result of the processing at time t_2 produces a background spectrum in green and a foreground spectrum in red, where the SOI is the only significant source of energy present. The foreground spectra generated may then be used for energy detection and basic metadata extraction, e.g. emission duration and bandwidth estimation.

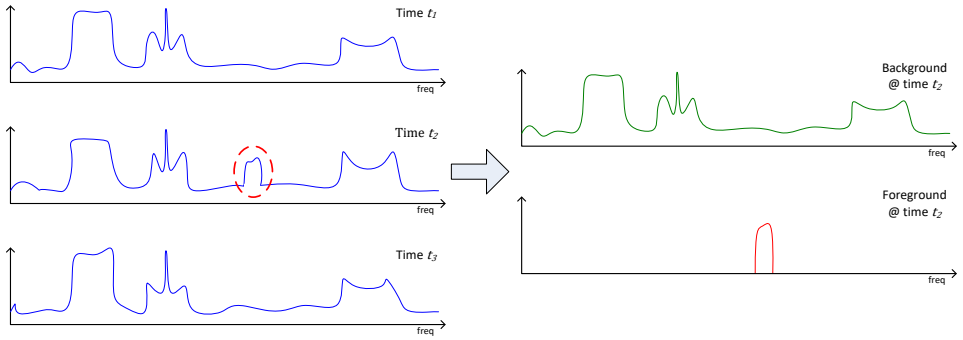

FIGURE 1. Decomposition of a radio spectrum into its background and foreground.

This paper is structured as follows: Section 2 contains a formal definition of the spectral decomposition problem. Section 3 then describes the data pre-processing required and the fundamental signal processing algorithm. Section 4 considers variants of the algorithm culminating in a version which is better suited to support the analysis of radio spectra. Experimental results are reported in Section 5 based on spectra recorded off-air in the 1 to 30 MHz band. The paper is concluded in Section 6.

2. Spectral decomposition problem definition

A wideband, digital receiver outputs a continuous stream of data samples. At time t a block of N samples is built up. The data block is multiplied by a window function, such as the Kaiser window, to reduce spectral leakage, and then a Fast Fourier Transform (FFT) converts the time domain data into the frequency domain. The Energy Spectral Density (ESD) is calculated by squaring the absolute values of the complex frequency domain data. The ESD values generated from a block of N samples at time t are expressed in decibels and stored as the elements of vector $\boldsymbol{m}(t)$. The conversion to decibels is important because it acts to compress the dynamic range so that low power, but still important, signals are not ignored by the spectral decomposition process.

A simple underlying model for $\boldsymbol{m}(t)$ is given by

$$\boldsymbol{m}(t) = \boldsymbol{s}(t) + \boldsymbol{l}(t) \tag{2.1}$$

where $\boldsymbol{s}(t)$ is a sparse vector containing the ESD of the fleeting SOIs, and $\boldsymbol{l}(t)$ is a dense vector which lies within a slowly changing, low-dimensional subspace that contains the background ESD. $\boldsymbol{l}(t)$ may be further modelled as

$$\boldsymbol{l}(t) = \boldsymbol{i}(t) + \boldsymbol{n}(t) \tag{2.2}$$

where $\boldsymbol{i}(t)$ represents the ESD of the background interference signals which are not of interest, and $\boldsymbol{n}(t)$ is unstructured noise (typically this is thermal noise from the radio receiver).

The justification for this model is as follows: the number of SOIs simultaneously transmitting is small, so the fraction of the total band occupied by SOIs is low; hence $\boldsymbol{s}(t)$ can reasonably be considered to be sparse. $\boldsymbol{l}(t)$ is at least relatively dense because the background signals are more numerous than the SOIs and they often transmit for long periods, so are very likely to be present in the spectrum. Furthermore, the subspace which contains $\boldsymbol{l}(t)$ is slowly changing and low-dimensional because the background emissions are not rapidly turning on and off, nor are they frequency agile; however, the sidebands of each background signal are continuously changing due to modulation at a rate commensurate with the signal bandwidth.

So the task at hand is to recover $\boldsymbol{s}$ from $\boldsymbol{m}$, with perhaps a lesser interest in also recovering $\boldsymbol{l}$.[2] Initially, this signal separation problem seems impossible to solve as the number of unknowns to recover $\boldsymbol{s}$ and $\boldsymbol{l}$ is $2N$ but there are only N measurements available in $\boldsymbol{m}$. However, under the conditions that $\boldsymbol{l}$ is not sparse and that the subspace containing $\boldsymbol{s}$ is not low-dimensional, then it can be proven that both $\boldsymbol{s}$ and $\boldsymbol{l}$ can be recovered precisely by convex optimization [1].

3. Compressive sensing for spectral decomposition

The family of algorithms investigated to recover the sparse component of the received signal ESD is called Recursive Projected Compressive Sensing (ReProCS) [7]. A key advantage of the approach is that it operates in a recursive manner on each new ESD vector $\boldsymbol{m}$. This means that it can respond rapidly to new energy without the latency incurred by batch processing data matrices built up from multiple ESD vectors over a longer period of time. The five fundamental stages of the ReProCS-NORST (Nearly Optimal Robust Subspace Tracking) [5] version of the algorithm are now explained.

3.1 Training

The first stage of the ReProCS algorithm comprises of training to calculate an initial basis for the subspace which contains $\boldsymbol{l}$ (i.e. the background signals). ReProCS-NORST specifies that a number of iterations of the non-convex AltProj algorithm are used during training [6]. A simpler approach is to directly apply Singular Value Decomposition (SVD) to a block of data, which was found to work well in practice.

A set of N_{tr} ESD vectors, which are assumed to contain no SOIs, are collected into a training data matrix, defined as $\mathbf{M}_{tr} = [\boldsymbol{m}(t), \boldsymbol{m}(t-1), \dots, \boldsymbol{m}(t-N_{tr})]$. The SVD is then applied

$$[\mathbf{B}, \mathbf{\Sigma}] \leftarrow \mathrm{SVD}(\mathbf{M}_{tr}) \tag{3.1}$$

where $\mathbf{B}$ contains the left singular vectors and $\mathbf{\Sigma}$ contains the singular values of $\mathbf{M}_{tr}$. The singular values are thresholded to account for a fixed percentage of the total energy (say 95%), and the corresponding columns of $\mathbf{B}$ are retained, i.e. rank reduction is employed to produce a low dimensional basis for $\mathbf{M}_{tr}$. As there are no foreground signals present in $\mathbf{M}_{tr}$, the reduced rank matrix $\mathbf{B}$ also forms a basis for the subspace of $\boldsymbol{l}$ (which is assumed to be dominated by the interference signals rather than the additive, white noise).

3.2 Orthogonal projection

In the first recursive processing step, $\boldsymbol{m}$ is projected into the space which is orthogonal to the subspace spanned by the columns of $\mathbf{B}$, i.e.

$$\boldsymbol{y} = \mathbf{P}\boldsymbol{m} = \mathbf{P}\boldsymbol{s} + \boldsymbol{v} \tag{3.2}$$

where $\mathbf{P} = \mathbf{P}_{\mathbf{B}}^{\perp} = \mathbf{I} - \mathbf{B}\mathbf{B}'$ is an orthogonal projection matrix[3] and $\boldsymbol{v} = \mathbf{P}\boldsymbol{l} = \mathbf{P}(\boldsymbol{i} + \boldsymbol{n}) \approx \mathbf{P}\boldsymbol{n}$.

If $\boldsymbol{s} = \mathbf{0}$ so that $\boldsymbol{m}$ only consists of background signals $\boldsymbol{l}$, then vector $\boldsymbol{y}$ will be comprised of noise values because $\boldsymbol{l}$ mostly exists within the span of $\mathbf{B}$, yet $\boldsymbol{y}$ only contains what is left after projecting $\boldsymbol{m}$ orthogonal to the span of $\mathbf{B}$.[4] In contrast, if $\boldsymbol{m}$ consists of both background and foreground signals ($\boldsymbol{l}$ and $\boldsymbol{s}$ respectively), then $\boldsymbol{y}$ can be expected to have some large elements because $\boldsymbol{s}$ does not lie within the span of $\mathbf{B}$. In other words, the action of $\mathbf{P}$ is to nullify most of the contribution of $\boldsymbol{l}$ so that $\boldsymbol{v}$ can be interpreted as noise.

---

[2] From now on the dependency of the signals on time t will be dropped where possible to simplify the presentation.

[3] We use the notation that $\mathbf{X}'$ is the transpose of $\mathbf{X}$.

[4] The components of the background $\boldsymbol{l}$ which lie outside the span of $\mathbf{B}$ are the thermal noise vector $\boldsymbol{n}$ plus any new components which occur due to interference signals $\boldsymbol{i}$ (slowly) changing over time.

3.3 Sparse recovery

From Equation 3.2, extracting $\boldsymbol{s}$ from $\boldsymbol{y}$ becomes a problem of sparse recovery in the presence of noise, which is a standard compressive sensing problem [2]. A sparse solution may be found by constrained ℓ_1 norm minimization,

$$\boldsymbol{s}_{cs} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{ subject to } \|\boldsymbol{y} - \mathbf{P}\boldsymbol{x}\|_2 \leq \xi \tag{3.3}$$

where $\xi = \mathbf{P}\boldsymbol{l}$ sets the required accuracy of the error constraint, which in turn enforces a good fit to the data. This optimization function is non-linear but it is convex, so a global minimum can be efficiently found.

The support of $\boldsymbol{s}$ is the set of indices at which the elements of $\boldsymbol{s}$ are non-zero, i.e. the frequency bins where foreground signal energy is present. We can calculate this support, denoted as T, by thresholding the elements of $\boldsymbol{s}_{cs}$:

$$T \leftarrow \{i : |\boldsymbol{s}_{cs,i}| > \omega_{supp}\} \tag{3.4}$$

where ω_{supp} is the threshold applied to decide between significant and non-significant elements of $\boldsymbol{s}_{cs}$.[5] The purpose of estimating T is that a refined version of $\boldsymbol{s}$ can now be found by least squares estimation on the estimated support, and setting it to zero everywhere else. This is expressed as

$$\boldsymbol{s}_T = (\mathbf{P}_T{}'\mathbf{P}_T)^{-1}\mathbf{P}_T'\boldsymbol{y} \text{ and } \boldsymbol{s}_{T^c} = 0 \tag{3.5}$$

where $\mathbf{P}_T$ is the sub-matrix of $\mathbf{P}$ that only contains the columns with indices in the set T, and T^c is the complement of T.

3.4 Low-dimensional vector estimation

With an estimate of $\boldsymbol{s}$ now available, Equation 2.1 can be rearranged so that

$$\boldsymbol{l} = \boldsymbol{m} - \boldsymbol{s} \tag{3.6}$$

Therefore, if $\boldsymbol{s}$ is accurately recovered then so is $\boldsymbol{l}$.

3.5 Subspace update

The three processing stages covered in Sections 3.2 to 3.4 can be repeated indefinitely to recursively estimate $\boldsymbol{s}$ and $\boldsymbol{l}$. However, if the background signals change over time then the subspace that they lie within will change. In this case, the basis matrix $\mathbf{B}$ needs to be updated as part of the recursive algorithm.

Changes in the subspace which contains $\boldsymbol{l}$ are detected by processing the last α estimates of $\boldsymbol{l}$. Let $\mathbf{L} = [\boldsymbol{l}(t), \boldsymbol{l}(t-1), \ldots, \boldsymbol{l}(t-\alpha)]$ and then project this matrix to be orthogonal to the span of $\mathbf{B}$,

$$\mathbf{D} = \mathbf{P}\mathbf{L} \tag{3.7}$$

If the largest singular value of $\mathbf{D}$ is calculated, then this can be inspected to decide whether the subspace of $\boldsymbol{l}$ has significantly changed. The rationale is that the singular value will be small if the subspace is unchanged because the columns of $\mathbf{L}$ all lie within the current subspace estimate. Hence, projecting $\mathbf{L}$ orthogonal to the current subspace estimate produces small entries in $\mathbf{D}$. Conversely, if the subspace has significantly changed then $\mathbf{D}$ will have larger elements, and its largest singular value will exceed a test threshold.

---

[5] Thresholding is necessary because although minimizing the ℓ_1 norm promotes sparsity in $\boldsymbol{s}$, it does not directly induce it; we are often left with some nonzero elements which are very small, so these may be neglected with little error by application of a threshold on their size. Alternatively, minimizing the ℓ_0 norm does directly penalize non-sparse solutions, thus avoiding the need for thresholding. Alas the ℓ_0 norm is not convex, so the optimization is no longer computationally tractable. Hence, the main reason for using the ℓ_1 norm is that it is a convex relaxation of the ℓ_0 norm.

If a subspace change has been detected, then the basis matrix **B** (which forms a spanning set for the subspace) is updated by

$$[\mathbf{B}, \mathbf{\Sigma}] \leftarrow \mathrm{SVD}(\mathbf{L}) \tag{3.8}$$

Here, **B** is equated with a subset of the left singular vectors which correspond to the dimensions that contain the majority of the total energy—calculated in a process similar to that described in Section 3.1. Note that the subspace update can occur even if foreground signals were present during the period spanned by the last α estimates of $\boldsymbol{l}$.

4 Algorithm variants

Apart from the initial training stage, ReProCS as covered in Section 3 is for the NORST variant which is described as being 'nearly optimal' [5]. In this section, a brief overview of the differences between a number of ReProCS variants is provided. Additionally, a new version is introduced which is better suited to the specific problem of decomposing radio spectra.

For all of the published ReProCS algorithm variants the orthogonal projection, sparse recovery and low-dimensional vector estimation stages are essentially the same. There are subtle differences in how the threshold used to enforce sparsity is calculated, but equivalent results can always be obtained. The main differences to note are in the subspace update stage where the basis matrix **B** is updated in response to changes in the background signals.

4.1 PracReProCS

PracReProCS is intended to be a practically usable modification of a theoretical algorithm [3]. One of its additions is a criterion used to detect whether the change in the support of $\boldsymbol{s}$ is fast or slow. If slow support change is detected then the compressive sensing optimization (Equation 3.3) is modified with the intent of producing a better estimate. For the radio spectrum decomposition problem, the presence of significant noise meant that fast support change was always detected, so making decisions based on the rate of change of support had no effect.

For PracReProCS, the detection of change in the basis matrix **B** did not seem to work when applied to actual radio receiver data. Change was always detected because the largest singular value of **D** is compared to the, invariably smaller, minimum singular value of the training data. Furthermore, during subspace update, **B** always increases in dimension by including new directions specified by singular vectors of **D**. This ultimately means that the rank of the subspace of $\boldsymbol{l}$ increases indefinitely, which is not helpful for a practical implementation.

4.2 s-ReProCS

Compared to PracReProCS, the so-called simple-ReProCS algorithm has a more successful subspace update stage [4]. Change in **B** is more reliably detected because the largest singular value of **D** is compared to a threshold calculated from the scaled minimum eigenvalue of the training data covariance matrix. Consequently, the scaling factor can be set to avoid an excessive frequency of subspace updates.

If a change in **B** is detected then additional directions are added to **B**, provided by the largest singular vector of **D**. Just one vector is added per update period consisting of α spectral estimates. During the last update period a full re-estimation of the basis **B** is calculated via a full SVD of **L**. The slow update over multiple update periods means that the algorithm is less responsive to sudden changes in **B**, which can occur if a background signal suddenly disappears.

4.3 ReProCS-NORST

The only significant difference between ReProCS-NORST [5] and s-ReProCS is that **B** is updated by performing a full SVD of **L** during a single update period. This means that changes in the background are more quickly assimilated into **B**.

4.4 RF-ReProCS

Starting with ReProCS-NORST it soon became apparent that it had some deficiencies when it was applied to spectral decomposition, so a modified version called Radio Frequency ReProCS (RF-ReProCS) was developed which is particularly appropriate for this problem. The first customised processing step was for the determination of the support of $\boldsymbol{s}$. Equation 3.4 was changed to be

$$T \leftarrow \{i : \boldsymbol{s}_{cs,i} > \omega_{supp}\} \tag{4.1}$$

because the data is in decibels, so the absolute value of the elements of $\boldsymbol{s}_{cs}$ should not be used. Furthermore, the support threshold was changed to be data driven,

$$\omega_{supp} = q\sqrt{\boldsymbol{m}'\boldsymbol{m}/N} \tag{4.2}$$

where $q \geq 1$ is a real scaling factor. This adaptive threshold improved the performance when the power levels of the signals present in the data vector change over time.

Another modification was to update the mean value of the background spectrum during the subspace update stage. This operation seems to be neglected in the other ReProCS variants.

Finally, ReProCS largely ignores the unstructured noise vector $\boldsymbol{n}$ during the processing. This may be reasonable for data output from high SNR video sensors, but it is not sensible for noisy radio spectra. A significant reduction in noise appearing in the foreground spectra is achieved by median filtering. Such a method operates over multiple data blocks, so it does increase the latency of the spectral decomposition process. Nonetheless, even operating over just 3 to 7 data blocks is worthwhile, and only has a small impact on short duration signal detection. Note that the shortest signal duration that can be reliably detected as a foreground signal is a function of the sample rate F_s, FFT length N, and window length of the median filter w_{median}:

$$\tau_{min} = \frac{N}{F_s}\left\lceil\frac{w_{median}}{2}\right\rceil \tag{4.3}$$

5. Experimental results

To provide an authentic challenge for the RF-ReProCS spectral decomposition algorithm, 1 MHz bandwidth data was recorded using a digital receiver operating in the 1 to 30 MHz band. To ensure that a short duration, foreground signal was present to be detected, test equipment was used to supplement the off-air signals. The generated signal was combined with the signal from a monopole antenna prior to feeding the receiver, as is shown in Figure 2.

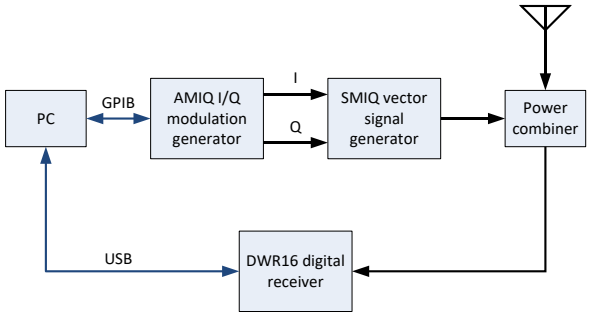

FIGURE 2. Data collection equipment.

During the data collection, the modulation generator was configured to produce 50 ms bursts of a random Quarternary Phase Shift Keyed (QPSK) signal, with a pulse repetition interval of 2 s. The spectra in Figure 3 are the result of processing data centred on 1.1 MHz in the middle of the busy medium wave broadcast band. Note that the FFT length was fixed at 1024 bins and a Kaiser window was used to reduce spectral leakage. The power level of the fleeting signal was

adjusted to produce the required SNR. Experiments were run over a range of SNR and the lower limit for successful operation was found to be 10 dB. Below 10 dB, SOIs are not fully attributed to the foreground, which results in missed detections. For the examples shown here, the SNR is approximately 20 dB.

The blue unprocessed spectrum contains numerous amplitude modulated signals as well as the QPSK signal centred on 675 kHz relative to the lower band edge. RF-ReProCS has correctly separated the SOI from the background signals despite the QPSK signal being very hard to spot by eye within the unprocessed spectrum. Also note that the additive noise is assigned to the background spectrum.

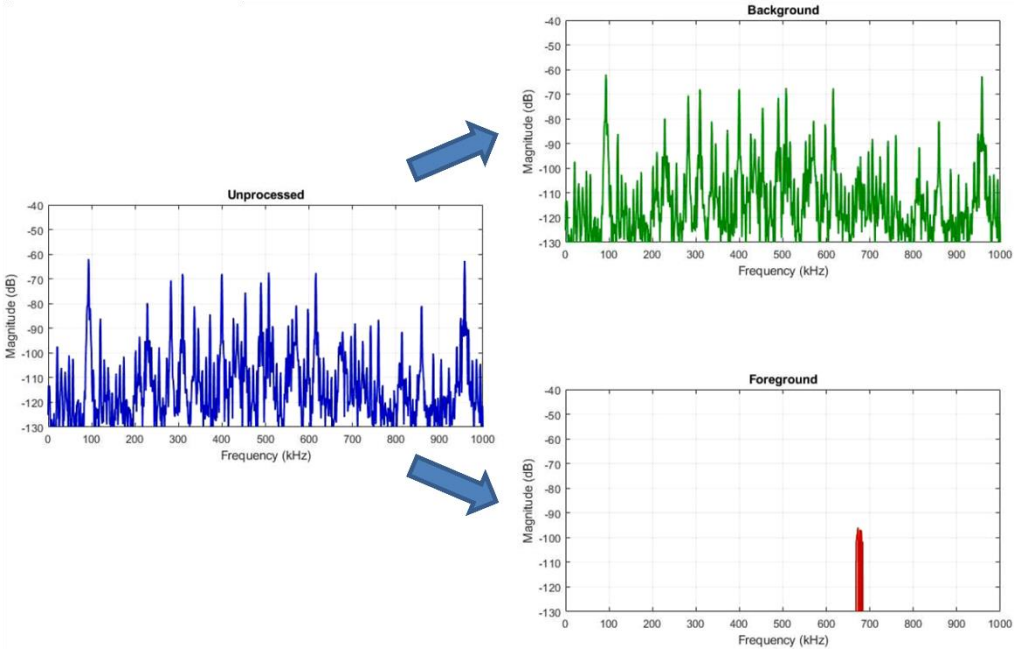

FIGURE 3. Spectra of the unprocessed data, and the extracted background and foreground signals for a very congested electromagnetic environment.

Spectrograms for the congested medium wave example are shown in Figure 4. The horizontal cyan line across the unprocessed data spectrogram marks the end of the training period used to estimate an initial basis for the subspace which contains the background signals. In the unprocessed data spectrogram the SOI is fairly obvious to the human eye; but it must be remembered that the spectrogram covers a period of 6 s, whereas the RF-ReProCS algorithm is working with individual rows of the spectrogram (about 1 ms of data) so that fleeting signals are detected with low latency. It is clear in the foreground spectrogram that there is hardly any breakthrough from the multitude of broadcast signals.

Figure 5 shows spectrograms for a situation where the background experiences significant time-variation. The receiver data is centred on 8.8 MHz, and the QPSK signal is positioned 550 kHz from the lower band edge. The foreground spectrogram shows the QPSK bursts together with several narrowband waveforms that turn on after the initial training period. The green horizontal lines define the α length window of estimates of $\boldsymbol{l}$ which are used to decide that the subspace in which $\boldsymbol{l}$ lies has changed. A subspace update is then instigated over the time period of the spectrogram demarked by blue horizontal lines.

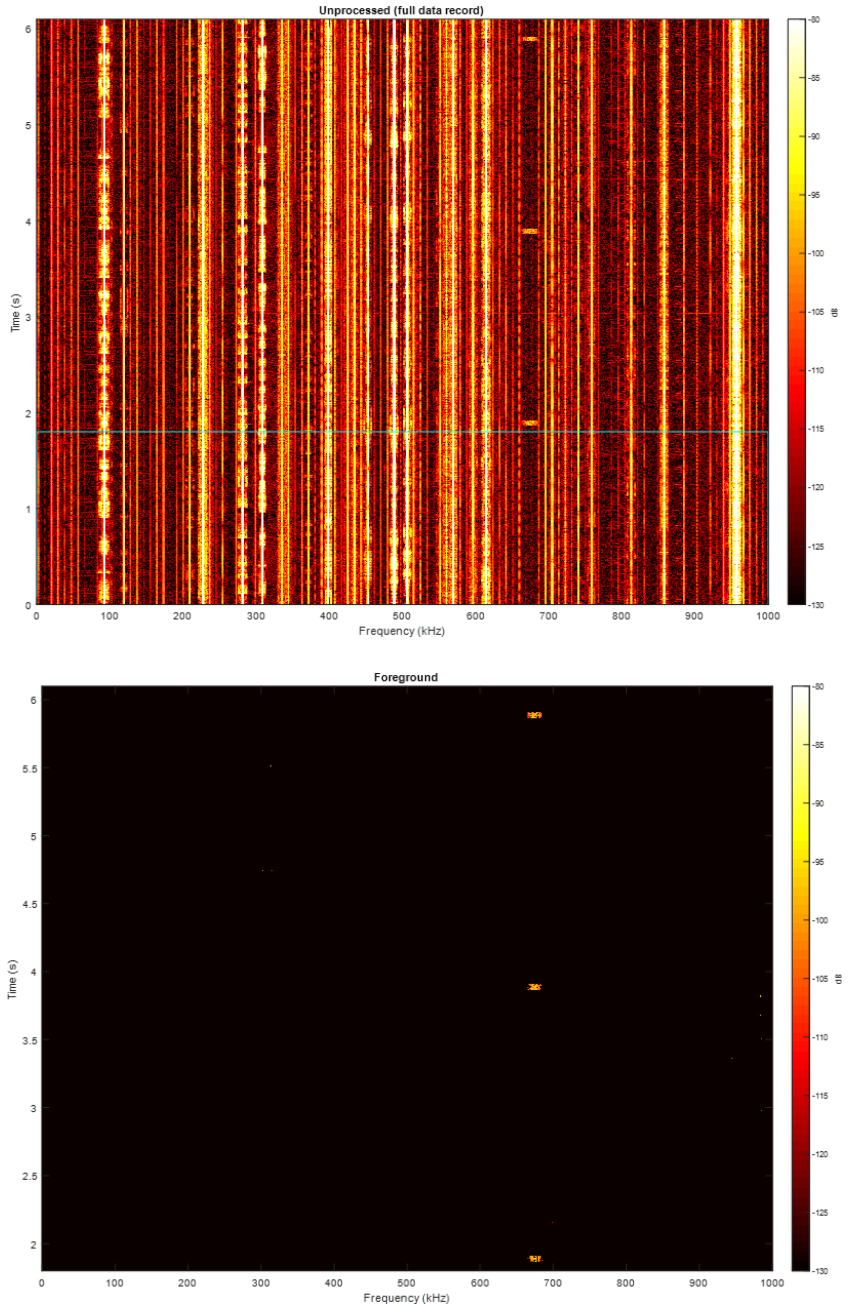

FIGURE 4. Spectrograms for the very congested electromagnetic environment example.

In Figure 5 it is seen that there exists foreground signals in addition to the QPSK SOI. This is valid because a number of off-air signals do appear after the initial training period is complete. However, there is also some unwanted breakthrough of the frequency shift keyed signal at 219 kHz into the foreground despite its presence during the initial training period. A subspace change is detected at 4.9 s and an update is instigated primarily because the strong pulsed signal at 517 kHz disappears. During the subspace update, the SOI is sufficiently removed from the span of basis matrix **B** so that it continues to be correctly placed in the foreground even though retraining of the subspace occurred when the SOI was present. This is not the case for the

weaker, narrowband signal at 643 kHz which is correctly placed in the foreground prior to the subspace update, but not afterwards. The reason for this behaviour is that weak foreground signals may not be completely removed from the low-dimensional subspace, so during the subspace update they are incorporated into the new basis, which has the consequence of transitioning them from the foreground into the background. Such behaviour only occurs for weak foreground signals which exist for the majority of the subspace update period, i.e. signals that are not fleeting.

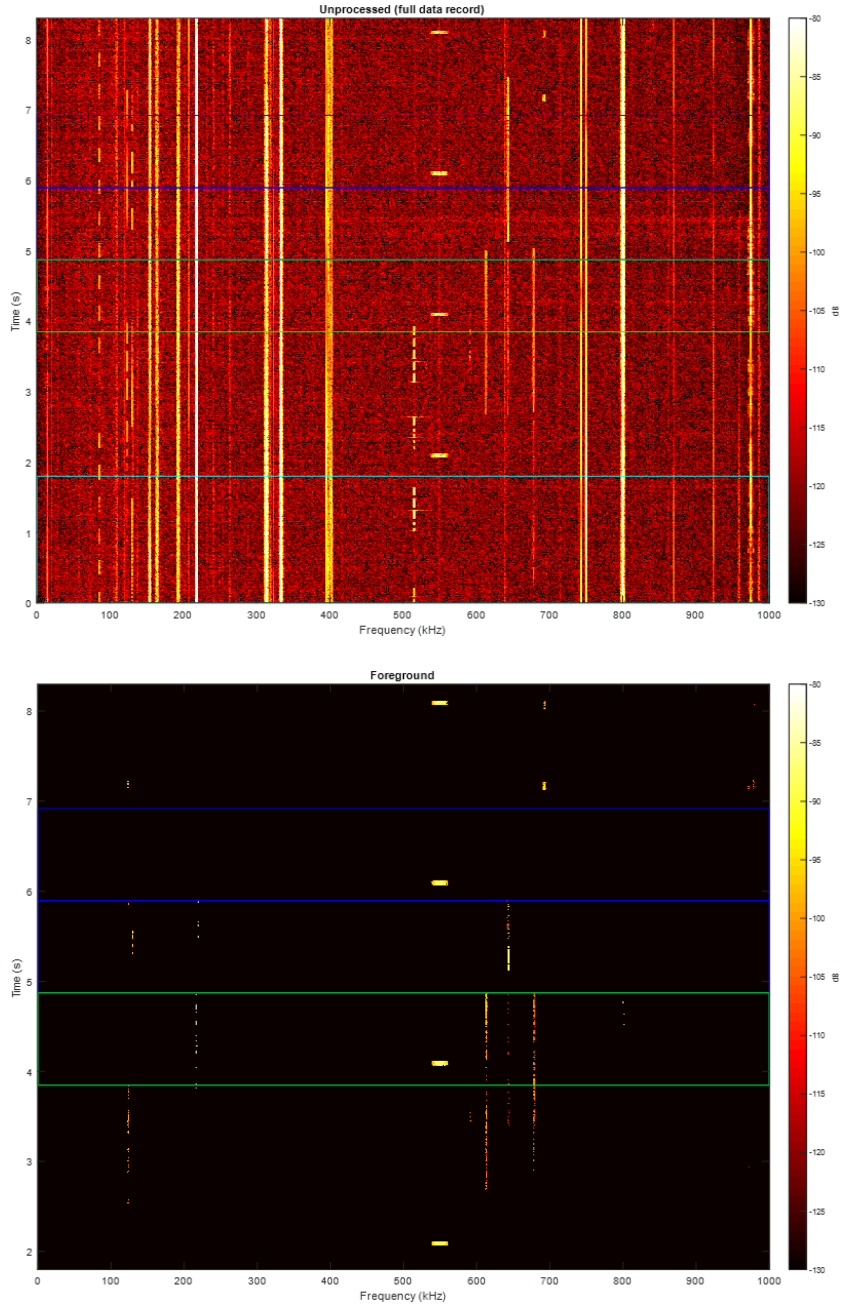

FIGURE 5. Spectrograms for data containing time-varying signals.

6. Conclusions

A method to decompose radio spectra into separate background and foreground spectra was developed based on a compressive sensing algorithm designed for video processing. The new algorithm, RF-ReProCS, operates on spectra converted to decibels so that low power signals are not ignored by default; has an adaptive threshold to determine support which improves performance when the signal power levels change over time; and incorporates median filtering to improve performance at low SNR. Experiments using off-air data within the 1 to 30 MHz band have confirmed the following advantages of the algorithm:

- Successful operation detecting short duration signals within the limit set by Equation 4.3.
- Robustness to the number and strength of background signals present.
- Operation when the foreground signals are not particularly sparse.
- Foreground signal noise reduction due to noise being assigned to the background.
- Ability to discern foreground signals which are close in frequency to background signals.
- Ability to adapt to changing background signals whilst still separating the spectrum into foreground and background.

A limitation of the approach is that operation at a SNR less than 10 dB is difficult. A suggestion is to combine the RF-ReProCS algorithm with beamforming to boost the effective SNR, using data collected by an array of antennas and receivers. Conventional beamforming can be used to steer high gain beams in different directions, but prior to signal detection there is no direction constraint available to guide this process. Instead, a blind beamforming approach such as one of those described in [8] would be suitable.

7. Acknowledgement

The author would like to express appreciation to the Defence Science Technology Laboratory (Dstl) which funded this work under contract number 4500038199.

REFERENCES

[1] E. J. CANDÈS, X. LI, Y. MA, J. WRIGHT, "Robust principal component analysis?", *J. ACM*, vol. 58, no.3, Dec. 2011.

[2] S. S. CHEN, D. L. DONOHO, M. A. SAUNDERS, "Atomic decomposition by basis pursuit", *SIAM J. Sci. Computing*, vol. 20, iss. 1, pp. 33-61, 1998.

[3] H. GUO, C. QIU, N. VASWANI, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum", *IEEE Trans. Sig. Proc.*, vol. 62, no. 16, pp. 4284-4297, Aug. 2014.

[4] P. NARAYANAMURTHY, N. VASWANI, "Provable dynamic robust PCA or robust subspace tracking", *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1547-1577, Mar. 2019.

[5] P. NARAYANAMURTHY, N. VASWANI, "Nearly optimal robust subspace tracking", in *Proc. Int. Conf. Machine Learning*, pp. 3701-3709, 2018.

[6] P. NETRAPALLI ET AL., "Non-convex robust PCA", in *Proc. Adv. Neural Inf. Processing Systems*, 2014.

[7] C. QIU, N. VASWANI, B. LOIS, L. HOGBEN, "Recursive robust PCA or recursive sparse recovery in large but structured noise", *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 5007-5039, Aug. 2014.

[8] D. J. SADLER, "Method and apparatus for signal detection", *UK patent GB2494204(B)*, 24 May 2017.