Feature Extraction of Galvanic Skin Responses by Non-Negative Sparse Deconvolution

Article *in* IEEE Journal of Biomedical and Health Informatics · December 2017

DOI: 10.1109/JBHI.2017.2780252

3 authors:

Francisco Hernando Gallego
University Carlos III de Madrid
3 PUBLICATIONS **64** CITATIONS
SEE PROFILE

David Luengo
Universidad Politécnica de Madrid
148 PUBLICATIONS **2,535** CITATIONS
SEE PROFILE

Antonio Artés Rodríguez
University Carlos III de Madrid
208 PUBLICATIONS **2,848** CITATIONS
SEE PROFILE

Feature Extraction of Galvanic Skin Responses by Non-Negative Sparse Deconvolution

Francisco Hernando-Gallego,[1] David Luengo,[2] *Member, IEEE*, and Antonio Artés-Rodríguez,[1] *Senior Member, IEEE*

***Abstract*—Wearable sensors are increasingly taking part in daily activities, not only because of the recent society health concern, but also due to their relevance in the medical industry. In this paper, a Galvanic Skin Response (GSR) extraction technique has been developed in order to interpret Electrodermal Activity (EDA) records, which can be useful both for ambulatory and health applications. The core of the proposed approach is a novel feature extraction scheme that is based on a non-negative sparse deconvolution of the observed GSR signals. Unlike previous approaches, the resulting SparsEDA algorithm is fast (immediately extracting the skin conductance level and response), efficient (being able to work with any sampling rate and signal length), and highly interpretable (due to the sparsity of the extracted phasic component of the GSR). Results on real data from 100 different subjects confirm the good performance of the method, which has been released through a free web-based code repository.**

***Index Terms*—Electro Dermal Activity (EDA), Non-negative deconvolution, Galvanic Skin Response (GSR), Sparse approximation, Sympathetic Nervous System (SNS), Wearable sensors**

I. Introduction

THE term Galvanic Skin Response (GSR) refers to changes in the electrical properties on the surface of the skin in response to sweat secretions [1]. These secretions are due to an increment in sudomotor innervation, caused by the Sympathetic Nervous System (SNS), that results in changes in the GSR signals as the body responds to different daily circumstances: stress, temperature, anxiety, exertion situations, etc. [2]. Hence, the sympathetic activity can be measured by analysing the GSR signals, as already shown in [3]. For instance, this relation between SNS acts and the observed skin conductance [4], which determines the sympathetic innervation of the sweat glands [5], has been extensively used in stress detection applications (e.g., see [6]–[9]).

GSR signals carry information about SNS activity, but are also influenced by other factors, like temperature changes or sweating due to aerobic exercise [10]. The challenge in analyzing them is thus to develop a method which is able to extract only SNS activity symptoms while avoiding other unrelated components. Indeed, GSR signals, which are usually denoted as $s(t)$, can be expressed as a sum of two components [1]:

- The *tonic component*, $s_\ell(t)$, or Skin Conductance Level (SCL), which is a slow changing signal. The SCL is related to several non-SNS activity factors but also to the level of attention of the subject, even in the absence of instantaneous stimuli.
- The *phasic component*, $s_p(t)$, also called Skin Conductance Response (SCR), which is the reaction to sporadic SNS stimuli. The SCR, which is superimposed on top of the tonic component, includes higher frequency components and appears only within specific time windows whose length typically lasts from one to five seconds [1].

Furthermore, SCRs can be modeled as the standard linear convolution between a sudomotor SNS innervation, $d_p(t)$, that corresponds to the *non-negative* unknown *sparse* driver that causes the observable skin conductance response, and the response triggered by that driver, $r(t)$. Hence, GSR signals can be finally decomposed as [11],

$$s(t) = s_p(t) + s_\ell(t) = d_p(t) * r(t) + s_\ell(t). \qquad (1)$$

GSR signal processing models then focus on the estimation of the SCR and SCL components in the absence of other factors.

Alexander et al. [12] were the first to introduce a decomposition algorithm based on the model of Eq. (1). Their approach estimates first the SCL contribution, which is subtracted from $s(t)$, and then reconstructs the SCR signal using an iterative inverse filter deconvolution method. However, this method leads to a non-sparse driver that can present negative impulses, which are not physiologically interpretable. Moreover, it is very slow, thus preventing its application for long time registers or on-line signal extraction.

Benedek et al. [13] addressed the decomposition using a different signal model and considering a non-negative deconvolution scheme based on Gauss elimination to avoid negative SCRs. After estimating the SCL and subtracting it from $s(t)$, this approach obtains an initial estimation of the SCR through a Gauss elimination deconvolution. The negative components of the SCR are then removed by introducing an arbitrary waveform that is fitted by minimizing the error with respect to the observed signal. Unfortunately, this method produces a noisy driver which is not sparse, and the waveforms introduced to force the non-negativity are not physiologically interpretable.

In another work, Benedek et al. [14] proposed an alternative non-negative decomposition based on the model of Eq. (1) that shows common ground with the method of [12]. It is based on a spectral division deconvolution with a Gaussian window (after removing the estimated SCL component again), and the

[1] F. Hernando-Gallego and A. Artés-Rodríguez are with the Department of Signal Theory and Communications (Universidad Carlos III de Madrid) & Gregorio Marañón Health Research Institute, e-mails:{fhernando,antonio}@tsc.uc3m.es

[2] D. Luengo is with the Department of Signal Theory and Communications (Universidad Politécnica de Madrid), e-mail:david.luengo@upm.es

SCR detection is performed by searching for the zeros in the first derivative of the driver. This approach yields an individual estimate of the typical SCR shape through optimization, but the estimated driver is still non-sparse and the computation is slow (as shown in the simulations).

Greco et al. [15] proposed a non-negative sparse deconvolution based on cubic B-splines for the SCR component, an ARMA model for the sudomotor SNS innervation, and an additive white Gaussian noise term. Although the resulting convex optimization problem can be solved, the solution obtained is still not sparse (values close to zero, but not exactly equal to zero, are obtained) and the interpretability is not improved w.r.t. previous approaches. They introduce the joint estimation of the SCL and SCR components, but this method is still not fast enough for on-line application. Another sparse deconvolution technique has been introduced very recently in [16]. This approach recovers a truly sparse driver and takes into account potential discontinuities in the SCL due to motion artifacts. However, they assume that the length and shape of the response $r(t)$ are known, they do not enforce the separation between SCR events that typically occurs due to physiological reasons, and the resulting algorithm is not directly applicable for on-line extraction of SNS information.

In summary, the sparse nature of the driver $d_p(t)$ that triggers the SCR response has not been fully exploited by previous methods. Consequently, the interpretability of the decomposition obtained is limited, since true SCR events are difficult to locate in the driver and artificial signals that have no physiological interpretation are introduced by some methods (e.g., [13]) to reduce the error of the model. Furthermore, none of these approaches is able to provide the real-time results required for on-line operation. This work addresses all these issues, developing a novel non-negative sparse deconvolution method (SparsEDA), that introduces the following main contributions:

- Multi-scale analysis that addresses the variable time width of the SCR impulses by using an *overcomplete dictionary* that includes responses, $r(t)$, with different widths and selects the appropriate width for each driver's impulse automatically.
- Exploitation of the sparsity of the SCR component (in order to obtain an interpretable decomposition) by formulating the estimation of $d_p(t)$ as a *sparse inference* problem.
- Fast and efficient solution of the resulting optimization problem, thus allowing for on-line extraction of the SNS information.
- Fully automated implementation of the algorithm in Matlab (released through a free web-based repository) that requires only the selection of two easily interpretable parameters by the user.

The paper is organized as follows. First of all, Section II describes the core of the SparsEDA algorithm: the sparse non-negative deconvolution model used. This includes the description of the discrete-time equivalent of Eq. (1) in Section II-A; the multi-scale analysis to account for the variable width of the impulses in Section II-B; the approach used to model the SCL in Section II-C; the optimization problem for the joint estimation of the SCL and SCR components in Section II-D; and the post-processing stage in Section II-E. Then, Section III addresses three issues that are essential to obtain a robust and efficient implementation of the SparsEDA method: the preprocessing stage (Section III-A); the continuous mode of operation for on-line signal recovery (Section III-B); and the feature extraction (Section III-C). Finally, Section IV validates the method on 100 signals from 100 different subjects and Section V provides some concluding remarks.

II. Sparse Non-negative Driver Model

A. Discrete-Time Model

The discrete-time model that we consider in the sequel is the following:[1]

$$s[n] = s_\ell[n] + d_p[n] * r[n] + w[n]. \tag{2}$$

where $n = 0, \ldots, N-1$ with N denoting the total number of samples available; $s_\ell[n]$, $d_p[n]$ and $r[n]$ are obtained through uniform sampling of $s_p(t)$, $d_p(t)$ and $r(t)$ in Eq. (1), with a sampling frequency $f_s = 1/T_s$ Hz; and $w[n]$ is an additive white Gaussian noise (AWGN) term that takes into account both measurement noise and discretization error.

For finite-length sequences, Eq. (2) can be expressed more compactly in matrix form as

$$\mathbf{s} = \mathbf{s}_\ell + \mathbf{R}\mathbf{d}_p + \mathbf{w}. \tag{3}$$

where $\mathbf{s}_\ell = [s_\ell[0], \ldots, s_\ell[N-1]]^T$, $\mathbf{R}$ is an $N \times N$ Toeplitz matrix, $\mathbf{d}_p = [d_p[0], \ldots, d_p[N-1]]^T$ is an $N \times 1$ sparse non-negative vector,[2] and $\mathbf{w} = [w[0], \ldots, w[N-1]]^T$ is the noise vector. Since $r[n] = 0$ when $n < 0$ or $n > M-1$ (see Section II-B),

$$\mathbf{R} = \begin{bmatrix} r[0] & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ r[1] & r[0] & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & & & \ddots & & \\ r[M-2] & r[M-3] & \cdots & r[0] & 0 & \cdots & 0 & 0 \\ r[M-1] & r[M-2] & \cdots & r[1] & r[0] & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \ddots & & \\ 0 & 0 & \cdots & 0 & 0 & \cdots & r[0] & 0 \\ 0 & 0 & \cdots & 0 & 0 & \cdots & r[1] & r[0] \end{bmatrix}.$$

Our global aim is inferring both $d_p[n]$ and $s_\ell[n]$ jointly when $r[n]$ (and thus $\mathbf{R}$) is unknown. In order to achieve this goal, we describe first how to approximate $\mathbf{R}$ in Section II-B and then how to model $s_\ell[n]$ in Section II-C.

[1]Note that (2) is not the discrete-time equivalent of (1) (as obtained by applying the bilinear transformation [17]), but its sampled version. However, it is a standard model [11], [12], [14], [16].

[2]The vector $\mathbf{d}_p$ is sparse, since its number of non-null elements (indicated by its L_0 pseudo-norm, $\|\mathbf{d}_p\|_0$) is small compared to its length, i.e., $\|\mathbf{d}_p\|_0 = L \ll N$ [18]. Note that a universally accepted threshold to define a vector as sparse does not exist, but we may consider that a vector is sparse when it contains less than 10 % of non-null elements (i.e., $L/N < 0.1$).

B. SCR Model: Multi-scale Analysis

We assume that the sudomotor nerve activity can be described by a biexponential function [12] regarding the specific response triggered by the driver:

$$r(t) = e^{-t/\tau_2} - e^{-t/\tau_1}, \qquad \text{for} \quad t \geq 0. \tag{4}$$

According to [12], the optimum performance in their experiments is obtained by setting $\tau_2 = 0.75$ and $\tau_1 = 2$. Therefore, we will use these values in the sequel. Another assumption made in [12] is that the duration of a response varies between one and five seconds. Thus, if τ_1 and τ_2 are fixed, in order to construct several waveforms with different time scales we have to use different sampling periods when discretizing $r(t)$. This leads to an overcomplete dictionary, which is a common approach to circumvent the scale problem in sparse inference methods [18]. Using Q different sampling periods, we have

$$\mathbf{R}_{SCR} = [\mathbf{R}_1,\ \mathbf{R}_2,\ \ldots,\ \mathbf{R}_Q], \tag{5}$$

where $\mathbf{R}_{SCR}$ is an $N \times NQ$ matrix, and the $\mathbf{R}_q$ $(q = 1, \ldots, Q)$ are a collection of $N \times N$ matrices (the elements of the dictionary) constructed using a predefined set of waveforms $(r_1[n],\ \ldots,\ r_Q[n])$ with different scales. If these waveforms are carefully chosen, they will be able to provide a very good approximation of the SCR, even if its scale changes over time. The selection of the element used at each time instant is performed by the $NQ \times 1$ extended vector,

$$\mathbf{d}_{SCR} = [\mathbf{d}_1^T,\ \mathbf{d}_2^T,\ \ldots,\ \mathbf{d}_Q^T]^T, \tag{6}$$

which is still a sparse vector with $\|\mathbf{d}_p\|_0 \leq \|\mathbf{d}_{SCR}\|_0 \ll NQ$. Hence, the signal model becomes

$$\mathbf{s} = \mathbf{s}_\ell + \mathbf{R}_{SCR}\ \mathbf{d}_{SCR} + \mathbf{w}. \tag{7}$$

We propose using $Q = 5$ waveforms,

$$r_i[n] = e^{-nT_s/\tau_{2i}} - e^{-nT_s/\tau_{1i}},$$

for $n = 0, \ldots, M-1$ and $i = 1, \ldots, 5$, with $\tau_{1i} = k_i\tau_1$, $\tau_{2i} = k_i\tau_2$, $k_i \in \{\frac{1}{2},\ \frac{3}{4},\ 1,\ \frac{5}{4},\ \frac{3}{2}\}$, and $M = 10/T_s$ (i.e., M is the number of samples obtained in a time interval of 10 seconds using a sampling period T_s). Therefore, the SCR matrix finally becomes:

$$\mathbf{R}_{SCR} = [\mathbf{R}_{T_1},\ \mathbf{R}_{T_2},\ \mathbf{R}_{T_3},\ \mathbf{R}_{T_4},\ \mathbf{R}_{T_5}], \tag{8}$$

where $\mathbf{R}_{T_i}$ is constructed using $r_i[n]$ for $i = 1, \ldots, 5$. Note that this matrix allows us to cater both for short time-scale processes (e.g., 1-2 seconds) and long time-constant processes (e.g., 3-5 seconds). Furthermore, since we construct the periods $T_1, \ldots, T_5$ as a function of T_s, we are also able to deal with any sampling period automatically. Fig. 1 shows the shape of the five different SCR waveforms used to construct $\mathbf{R}_{SCR}$. Note that all of them correspond to an SNS occurring at the same time instant $t = 0$ (i.e., without any delay), although their effective lengths and the positions of the peaks of the SCR responses are different.[3]

[3]The peak of the SCR response is located $\frac{6\ln(8/3)}{5}k_i$ seconds after the SCR event that triggered the response. For the $k_i \in \{\frac{1}{2},\ \frac{3}{4},\ 1,\ \frac{5}{4},\ \frac{3}{2}\}$ used in Fig. 1 this corresponds approximately to $\{0.5885,\ 0.8827,\ 1.1770,\ 1.4712,\ 1.7655\}$ seconds, respectively.

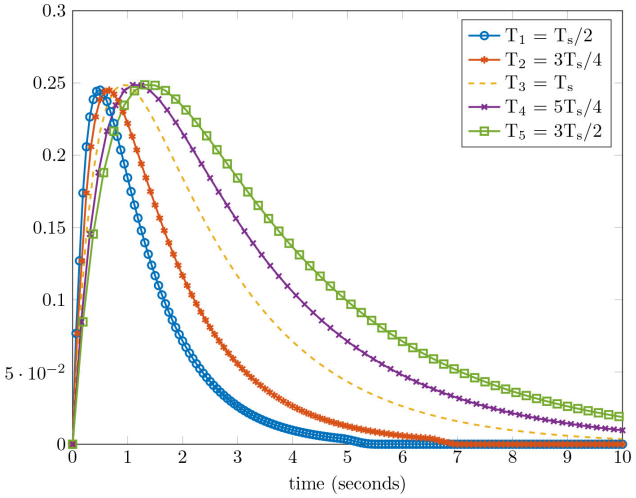

Fig. 1. Multi-Scale SCR waveforms $(r_1[n],\ \ldots,\ r_5[n])$, obtained discretizing $r(t)$ in Eq. (4) using five different periods: $T_1 = T_s/2$, $T_2 = 3T_s/4$, $T_3 = T_s$, $T_4 = 5T_s/4$, and $T_5 = 3T_s/2$. All of them correspond to an SNS occurring at the same time instant $t = 0$.

C. SCL Model: Taylor Series Expansion

In order to approximate the SCL in a simple and efficient way, we propose to use a first order Taylor series expansion:

$$\mathbf{s}_\ell = [\mathbf{1}/||\mathbf{1}||_2\ \ \boldsymbol{\ell}/||\boldsymbol{\ell}||_2\ \ -\boldsymbol{\ell}/||\boldsymbol{\ell}||_2]\ \mathbf{d}_{SCL} = \mathbf{R}_{SCL}\mathbf{d}_{SCL}, \tag{9}$$

where $\mathbf{1} = [1,\ \ldots,\ 1]^T$ and $\boldsymbol{\ell} = [0,\ 1,\ \ldots,\ N-1]^T$ are constant and linearly increasing $N \times 1$ column vectors, respectively; $\|\cdot\|_2$ denotes the L_2 norm, so $||\mathbf{1}||_2 = \sqrt{N}$ for instance; $\mathbf{R}_{SCL}$ is the $N \times 3$ matrix built using those column vectors; and $\mathbf{d}_{SCL}$ is a non-negative coefficients vector.

D. Joint SCL and SCR estimation

The standard approaches to GSR signal analysis in the literature are based on the sequential extraction of $s_\ell(t)$ and $s_p(t)$. On the one hand, $s_\ell(t)$ can be estimated as an average of $s(t)$ over short intervals (10 to 100 seconds [12], [13]), and then subtracted from $s(t)$ in order to obtain $s_p(t)$. Alternatively, $s_p(t)$ can be obtained from $s(t)$ through a high pass filter [19], and then, if desired, $s_\ell(t)$ can be obtained again by subtraction. Either way, as pointed out in [14], this sequential extraction tends to underestimate the SCR component.

Here, we propose to estimate both the SCL and the SCR components simultaneously, exploiting the fact that $\mathbf{R}_{SCL}$ and $\mathbf{R}_{SCR}$ can be combined into a single $N \times (NQ+3)$ matrix: $\mathbf{R}_T = [\mathbf{R}_{SCL}\ \mathbf{R}_{SCR}]$. Hence, we can rewrite Eq. (7) more compactly as

$$\mathbf{s} = \mathbf{R}_T\mathbf{d}_T + \mathbf{w}, \tag{10}$$

where $\mathbf{d}_T = [\mathbf{d}_{SCL}^T\ \mathbf{d}_{SCR}^T]^T \in \mathbb{R}^{NQ+3}$. This is the joint SCL-SCR model that we use in the sequel to perform the joint estimation of the SCL and SCR components.

The proposed solution is based on the following key characteristics of the skin conductance response [4], [20]:

- The driver, $d_p(t)$, represents sudomotor nerves activations, and thus it corresponds either to non-negative

deflections (during states of activity) or remains equal to zero otherwise.
- A single impulse corresponds to each SNS act [14] and triggers an SCR response that typically lasts from 1 to 5 seconds [14].
- The sudomotor impulses arrive as discrete and separate (i.e., non-overlapping) events, implying that $\mathbf{d}_{SCR}$ should be sparse in time.

These facts lead us to consider a *non-negative sparse driver*, with the constraint that $d_T(i) \geq 0$ for $1 \leq i \leq N$, which is characterized by a zero baseline and occasional (i.e., sparse) discrete positive impulses with a compact support. Hence, the estimation of $\mathbf{d}_T$ can be formulated as the following constrained optimization problem:

$$\hat{\mathbf{d}}_T = \underset{\mathbf{d}_T}{\arg\min} \quad \|\mathbf{s} - \mathbf{R}_T\mathbf{d}_T\|_2^2 \tag{11a}$$

$$\text{subject to} \quad d_T(i) \geq 0 \; \forall i \tag{11b}$$

$$\|\mathbf{d}_T\|_0 \ll NQ \tag{11c}$$

Eq. (11a) corresponds to the minimization of the mean squared error (MSE) between the available signal and the reconstructed one, Eq. (11b) is the non-negativity constraint, and Eq. (11c) is the sparsity constraint. Since the L_0 pseudo-norm is untractable from a mathematical point of view, the sparsity constraint in Eq. (11c) can be imposed using an L_1 regularization term, that leads to the following non-negative version of the LASSO (Least Absolute Shrinkage and Selection Operator) [21]:

$$\hat{\mathbf{d}}_T = \underset{\mathbf{d}}{\arg\min} \quad \|\mathbf{s} - \mathbf{R}_T\mathbf{d}_T\|_2^2 + \lambda\|\mathbf{d}_T\|_1 \tag{12a}$$

$$\text{subject to} \quad d_T(i) \geq 0 \; \forall i \tag{12b}$$

which can be solved efficiently using the LARS (Least-Angle Regression) algorithm [22].

Note that the regularization parameter, λ, controls the amount of sparsity in the reconstruction: the larger the value of λ the sparser the signal obtained. However, λ has no clear interpretation in terms of GSR signals, and thus it can be difficult for the user to find an appropriate value for it. Instead, since LARS is a greedy algorithm, we propose to replace the choice of λ by the selection of a stopping rule that can be set much more easily by users without a detailed technical knowledge of the algorithm. For instance, we stop the optimization when the residual of the GSR reconstruction is less than a pre-defined value (i.e., $\|\mathbf{s} - \mathbf{R}_T\hat{\mathbf{d}}_T\|_2^2 \leq \epsilon$) or a maximum number of iterations $K_{\max}$ have been performed. Furthermore, since a final post-processing stage is performed (see Section II-E) to remove redundant impulses, the user can simply set values of ϵ and $K_{\max}$ which ensure that all the potential impulses have been discovered by LARS (e.g., in the simulations we have used $\epsilon = 10^{-4}$ and $K_{\max} = 40$), and then control the degree of sparsity during the post-processing.

E. Post-Processing

The goal of the post-processing stage is to obtain a driver signal, $d_p(t)$, which is as sparse as possible. In order to achieve this goal, we propose to apply a greedy algorithm that eliminates weak impulses which are too close to stronger impulses, following a similar approach to [23], [24]. These impulses are included by LARS in order to decrease the MSE, but are useless from a physiological point of view and hinder the interpretability of the recovered signals. In summary, the proposed approach is the following:

1. Initialize the set of accepted impulses as the empty set: $\mathcal{A} = \emptyset$.
2. Sort the elements of $\mathbf{d}_{SCR}$ in descending order according to their L_1 norm. Hence, the resulting ordered vector, $\mathbf{d}_{ord}$, fulfills that
$$\|\mathbf{d}_{ord}(1)\|_1 \geq \|\mathbf{d}_{ord}(2)\|_1 \geq \cdots \geq \|\mathbf{d}_{ord}(N)\|_1.$$
Note that, since $\mathbf{d}_{SCR}$ is sparse, only the first $L_{SCR} \ll NQ$ elements of $\mathbf{d}_{ord}$ are different from zero.
3. For $i = 1, \ldots, L_{SCR}$:
   a) If the location of $\mathbf{d}_{ord}(i)$ is not within $T_{\min}$ seconds (i.e., $N_{\min} = T_{\min}f_s$ samples) of an existing impulse in $\mathcal{A}$, add it to the set of accepted impulses.
   b) Otherwise, discard it.
4. Discard the accepted impulses whose L_0 norm lies below the following threshold:
$$\gamma = \rho \max \|\mathbf{d}_{SCR}\|_1,$$
where $0 < \rho < 1$ is a user specified parameter that can be used to control the final sparsity of the solution obtained.

The minimum distance constraint enforced by the previous algorithm could also be included within the optimization problem, as shown by the Cross-Products LASSO algorithm [25]. However, this leads to a substantial increase in computational cost (since the resulting problem is not convex anymore), that would prevent the on-line implementation that we are seeking here.

III. Practical Implementation

A. Preprocessing

Continuous and unobtrusive measurement of GSR using wearable devices makes the signal collected vulnerable to several types of noise and artifacts. Artifacts can be generated from electronic noise or variation in the contact between the skin and the recording electrode caused by pressure, excessive movement or adjustment of the device. If these artifacts remain in the signal when it is analysed, they can easily be misinterpreted and skew the analysis; for example, they may be easily mistaken for an SCR [2].

In order to remove these artifacts, Sara Taylor et al. [2] developed a machine learning algorithm to detect automatically electrodermal activity (EDA) artifacts that will be used before applying our novel SparsEDA algorithm. This method takes segments of 5 seconds and classifies them either as artifacts or as valid GSR signals. In those slots that are classified as artifacts, the signal is replaced by a polynomial regression of order 1 between the last non-artifact point in previous slots and the first non-artifact sample in subsequent slots, and no further processing is performed.

Besides the artifact removal stage, the complete signal is resampled to 8 Hz if the sampling frequency, f_s, is higher than 8 Hz. This resampling does not have any influence on the signal's quality, since SCR waveforms can be perfectly represented using $f_s = 8$ Hz (or even $f_s = 4$ Hz), but allows us to reduce notably the computation time.

B. Continuous-mode Operation

The final goal of the SparsEDA method is being able to extract EDA-related features continuously using shorter signal sets of $N = W + M + 1 < L$ samples, where M is the length of the waveforms used to construct the SCR dictionary and L is the total number of samples available of the GSR signal. This continuous-mode operation allows us to deal with large signals of arbitrary length (since the approach is applied on segments of fixed length) and even to process signals on-line (i.e., as they are being acquired) in real-time or quasi-real-time.[4] In order to describe this mode of operation, let us set $M = 10/T_s$ (i.e., the duration of the SCR waveforms is 10 seconds), $W = 60/T_s$, and refer to the current slot being processed (whose duration is 70 seconds) as the *active set*.

First of all, a *naive implementation* of the SparsEDA algorithm for large/on-line signals is straightforward:

1. Extend the signal by adding M samples before the first one and W samples after the last one.[5]
2. Set the first N samples of the extended signal as the active set.
3. Apply the SparsEDA method described in Section II on the current active set, obtaining an estimate of its SCL and SCR components for the first W samples.
4. Keep shifting the active set by W samples and repeating the previous stage until all the samples in the original GSR signal have been processed.
5. Discard the SCL and SCR components corresponding to the samples added at the beginning and the end of the signal.

However, this naive approach does not ensure the continuity of the signal among the different segments for the SCL component and can lead to high MSE values. In order to promote the continuity, we propose to divide the active set into three continuous sets, composed of $\frac{1}{3}W$ samples each one, and to use a modified $\mathbf{R}_{SCL}$ matrix, $\mathbf{R}'_{SCL}$. This matrix, shown in Eq. (13), is composed of W rows (i.e., the length of the active set) and six columns that correspond, respectively, to the positive and negative versions of the SCL model for each of the three segments of the active set. In Eq. (13), the $\boldsymbol{\ell}_i$ are linearly increasing column vectors ($\boldsymbol{\ell}_1$ from 0 to 1 with a length $L_1 = W$, $\boldsymbol{\ell}_2$ from 0 to 2/3 with a length $L_2 = 2W/3$, and $\boldsymbol{\ell}_3$ from 0 to 2/3 with a length $L_3 = W/3$); $\bar{\boldsymbol{\ell}}_i = \boldsymbol{\ell}_i / \|\boldsymbol{\ell}_1\|_2$ are the normalized vectors (w.r.t. the L_2 norm of $\boldsymbol{\ell}_1$); and $\bar{\ell}_i(k)$ denotes the k-th element ($k = 1, 2, \ldots, L_i$) of $\bar{\boldsymbol{\ell}}_i$. This construction of the matrix enforces the continuity among those three segments, since a discontinuity results in an increase in the MSE. Hence, the *continuous-mode operation* SparsEDA algorithm proposed is finally:

1. Extend the signal by adding M samples before the first one and W samples after the last one.
2. Set the first N samples of the extended signal as the active set.
3. Subtract the value of the first sample,[6] and apply the SparsEDA method described in Section II on the current active set, using a matrix $\mathbf{R}'_T = [\mathbf{R}'_{SCL}\ \mathbf{R}_{SCR}]$, with $\mathbf{R}'_{SCL}$ given by (13) and $\mathbf{R}_{SCR}$ given by (8), thus obtaining an estimate of the SCL and SCR components for the first $W/3$ samples. Add back the value of the subtracted sample to obtain the final solution.
4. Keep shifting the active set by $W/3$ samples and repeating the previous stage until all the samples in the original GSR signal have been processed.
5. Discard the SCL and SCR components corresponding to the samples added at the beginning and the end of the signal.

Note that the difference w.r.t. the naive implementation lies in steps 3 and 4 (using a modified matrix R'_T and shifting the active set by $W/3$ samples instead of W).

The proposed SparsEDA algorithm has been made freely available (altogether with the results presented in Section IV, which are provided in a `.mat` file) through a well-known web-based code repository (https://github.com/fhernandogallego/sparsEDA) and can also be found in the first author's web-page (http://www.tsc.uc3m.es/~fhernando/Research.html). The code, which has been developed in MATLAB, is based on the implementation of the LASSO algorithm provided by [26].

C. Feature Extraction

The SparsEDA algorithm allows us to extract two GSR components (features) that should be interesting for SNS studies:

- **Tonic component (SCL):** The slope is related to the level of attention of the subject. If the patient is concentrated and/or involved in a task, an increasing slope should be observed [27]. Otherwise, the SCL slope should either decrease or remain constant.
- **Phasic component (SCR):** This is the indicator of SNS reactions. The proposed method allows us to extract both their locations and durations. Furthermore, the sparsity of the resulting driver enhances its interpretability, whereas the post-processing stage allows us to avoid false alarms.

An example of the application of the SparsEDA method is shown in Fig. 2. The input is a GSR signal of length $N = W + L + 1$ (in blue), and the results (features) are a sparse driver signal of length W (displayed as black impulses), the corresponding SCR component of length W (in green), and the

[4]The SparsEDA algorithm incurs in a fixed processing delay equal to the time shift used (W or $W/3$), but afterwards it is able to return the SCL and SCR values almost instantaneously

[5]These samples are required to ensure that the whole signal is properly processed and can take arbitrary values. For instance, we simply replicate the first and last samples in the signal M and W times, respectively, to perform this extension.

[6]Note that (13) does not contain the offset term for the Taylor series expansion used to model the SCL. Hence, we subtract the first sample of the active set to ensure that the line used to model the SCL starts at 0.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2017.2780252, IEEE Journal of Biomedical and Health Informatics

$$\mathbf{R}'_{SCL} = \begin{bmatrix} \bar{\ell}_1(1) & -\bar{\ell}_1(1) & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{\ell}_1(\frac{W}{3}) & -\bar{\ell}_1(\frac{W}{3}) & 0 & 0 & 0 & 0 \\ \bar{\ell}_1(\frac{W}{3}+1) & -\bar{\ell}_1(\frac{W}{3}+1) & \bar{\ell}_2(1) & -\bar{\ell}_2(1) & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{\ell}_1(\frac{2W}{3}) & -\bar{\ell}_1(\frac{2W}{3}) & \bar{\ell}_2(\frac{W}{3}) & -\bar{\ell}_2(\frac{W}{3}) & 0 & 0 \\ \bar{\ell}_1(\frac{2W}{3}+1) & -\bar{\ell}_1(\frac{2W}{3}+1) & \bar{\ell}_2(\frac{W}{3}+1) & -\bar{\ell}_2(\frac{W}{3}+1) & \bar{\ell}_3(1) & -\bar{\ell}_3(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \bar{\ell}_1(W) & -\bar{\ell}_1(W) & \bar{\ell}_2(\frac{2W}{3}) & -\bar{\ell}_2(\frac{2W}{3}) & \bar{\ell}_3(\frac{W}{3}) & -\bar{\ell}_3(\frac{W}{3}) \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{13}$$

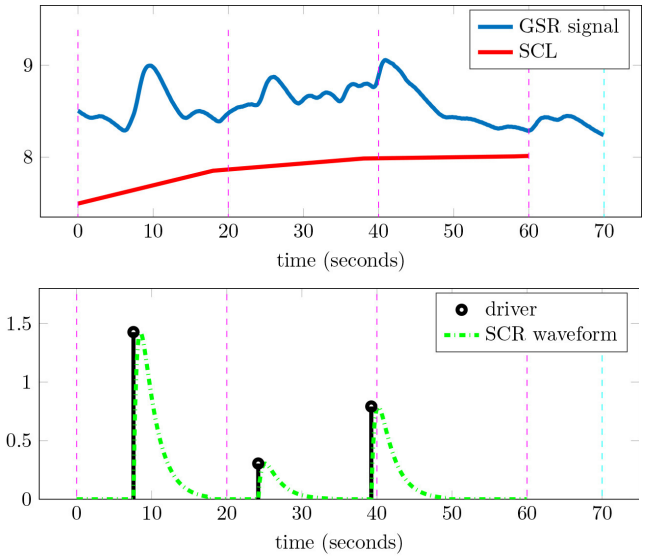

Fig. 2. Signal extraction example using SparsEDA. **(Top)** A GSR signal (blue) of duration equal to 70 seconds (i.e., length $L = 70/T_s$ samples), and the SCL signal (red) obtained by SparsEDA. **(Bottom)** Sparse driver obtained (black) and the SCR waveform for each mark (dash-dotted green).

low-frequency SCL components for each length $\frac{1}{3}W$ set (in red) with their associated slope. Purple vertical lines separate each $\frac{1}{3}W$ set, whereas the cyan line indicates the end of the signal.

IV. Results

First of all, we show the qualitative behavior of SparsEDA on a record acquired at $f_s = 16$ Hz with length equal to 400 seconds that is freely available at [28]. Fig. 3 compares the SCL and SCR components extracted by the novel SparsEDA algorithm (without decreasing the sampling rate) and two alternative approaches: the Continuous Decomposition Analysis (CDA) technique introduced in [14] (CDA Ledalab), and the convex approach proposed in [15] (cvxEDA). Regarding the SCL component, it can be seen that SparsEDA retrieves an SCL component which is very similar to the one returned by cvxEDA, and both of them lie below the SCL component obtained using CDA Ledalab. However, the main difference can be appreciated in the SCR component. On the one hand, the SCR signal returned by CDA Ledalab is not at all sparse and the driver's impulses are very difficult to locate. On the other hand, although cvxEDA returns a sparser SCR component, it still contains too many activations to be useful for the task of locating the driver's impulses and counting their number. Finally, SparsEDA returns a truly sparse SCR signal (with a degree of sparsity that can be easily controlled by the user, as described in Section II) that contains the main impulses of the driver and which is much more interpretable than the other two from a physiological point of view.

Then, in order to show the versatility of the proposed approach, we have compared the performance of the novel SparsEDA method with the same two algorithms as before: CDA Ledalab [14] and cvxEDA [15]. The simulations have been performed on a database composed of 100 GSR signals from 100 different patients acquired at several sampling rates from 4 Hz to 128 Hz, with different sensors and a wide range of signal lengths. 50 signals were recorded within the ES3 project [29], using Medicom MTD sensors [30] and following the procedure described in [31]. The rest were recorded in our laboratory using: the Empatica E4 sensor [32] (5 signals), Microsoft's Band 2 [33] (5 signals), Qsensor [34] (35 signals), and Shimmer3 [35] (5 signals). Regarding our SparsEDA algorithm, we have used the same parameters as before: $\epsilon = 10^{-4}$, $K_{\max} = 40$ iterations, $N_{\min} = \frac{5}{4}f_s$ samples, and $\rho = 0.025$. All the signals have been preprocessed using the web-based method proposed in [2], segmenting them into slots of 5 seconds that correspond either to a valid GSR signal or artifact/noise, and replacing the slots that were labelled as artifacts by a linear regression. After this preprocessing stage, each signal was processed using the three aforementioned algorithms, extracting the SCR and SCL components.

Table I summarizes the characteristics of the six groups of data used in the simulations (number of signals, sampling rate, type of sensor used and duration [mean and standard deviation (std.)]), altogether with the results obtained: the relative MSE for SparsEDA (mean and std.) and the computation time for the three algorithms tested (mean and std.). Note that sampling rates larger than 8 Hz has been re-sampled to 8 Hz for all of the methods. A detailed analysis of the results obtained is

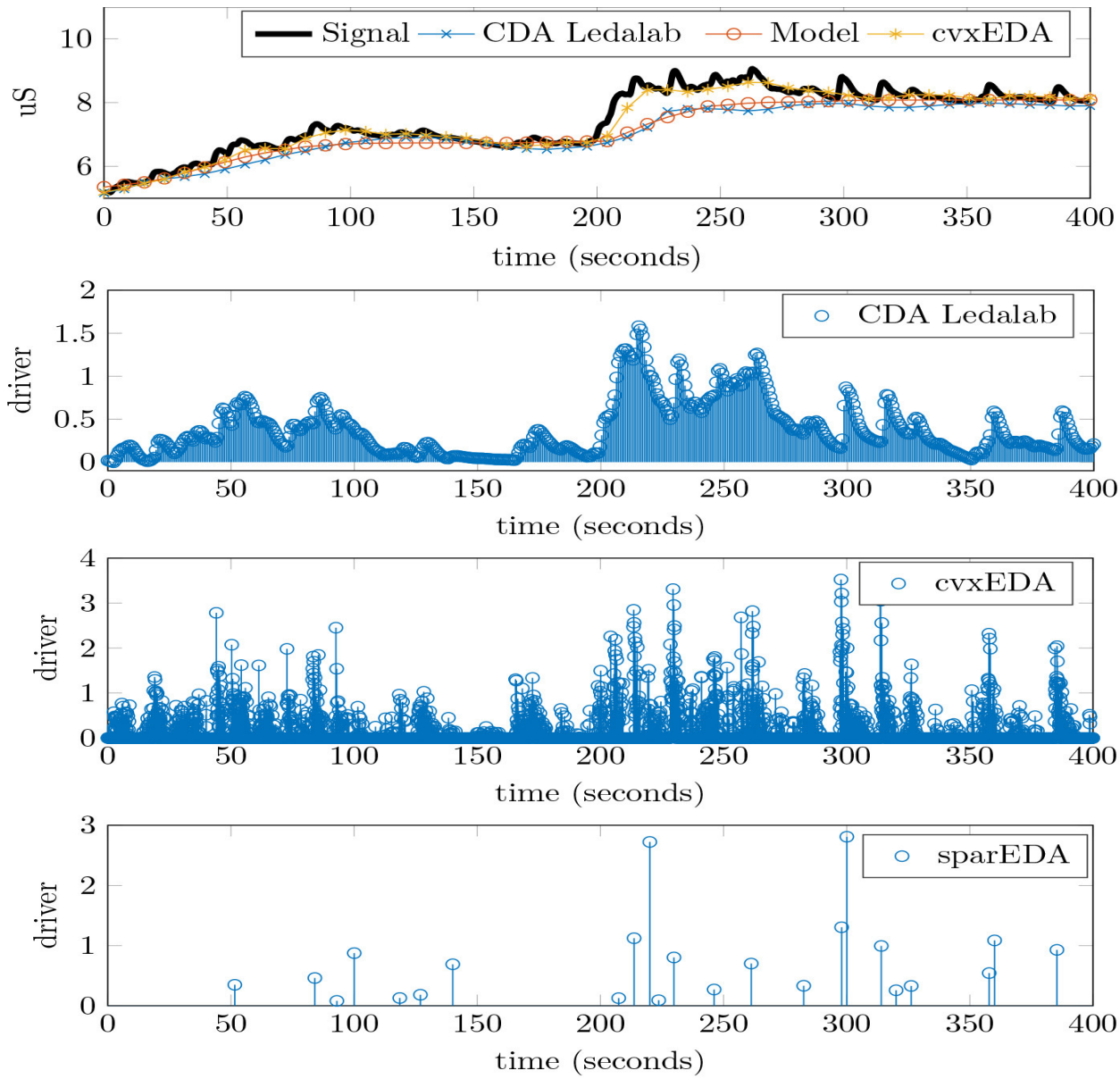

Fig. 3. Signal processing for GSR signal feature extraction using a sampling rate $f_s = 16$ Hz. (a) Black line representing the GSR signal and several lines with markers showing different SCL approximations: CDA Ledalab (blue crosses), SparsEDA (red circles) and cvxEDA (yellow stars). (b) CDA Ledalab driver obtained using 4 iterations of the optimization tools. (c) Driver obtained using cvxEDA. (d) Driver obtained using SparsEDA with $\epsilon = 10^{-4}$, $K_{\max} = 40$ iterations, $N_{\min} = \frac{5}{4} f_s = 20$ samples, and $\rho = 0.025$.

provided in the sequel:

- Fig. 4 displays the MSE obtained by SparsEDA and cvxEDA for each record. Let us remark that the main goal of SparsEDA is obtaining an interpretable SCL/SCR decomposition in a small amount of time (i.e., attaining a good balance between computation time and MSE), not fitting the SCL and SCR components in order to obtain a zero remainder. However, very good values of MSE are attained: from a minimum value of -83.29 dB to a maximum value of -29.55 dB, with an average MSE equal to -48.50 dB. Although cvxEDA attains lower MSE values (from -111.26 dB to 5.47 dB, with -50.95 dB on average), this is hindered by the much lower interpretability of the resulting SCR signal (as shown in Fig. 3 and discussed below).
- The MSE obtained by CDA Ledalab is not included in Fig. 4 because it is fitted to zero by using a non-sparse SCR component (which is not physiologically interpretable) that contains all the elements of the signal that do not belong to the SCL.
- The computation time is summarized for each method in Table I, showing that SparsEDA is always faster than cvxEDA, and both of them are much faster than CDA Ledalab. Fig. 5 delves deeper into this important feature, showing the ratio between the computation time (in seconds) and the duration of the signal (in minutes). The average values of the ratios, which are also displayed, are 1.327 for CDA Ledalab, 0.088 for cvxEDA and 0.012 for SparsEDA (i.e., on average SparsEDA is 7.42 times faster than cvxEDA and 111.58 times faster than

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2017.2780252, IEEE Journal of Biomedical and Health Informatics

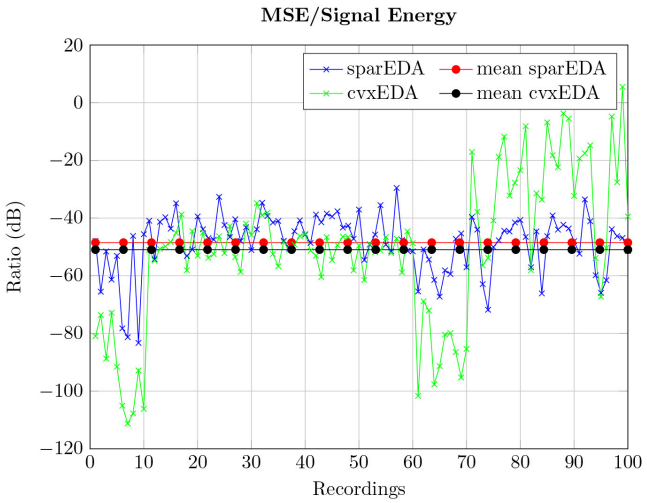

Fig. 4. Ratio (dB) between the MSE and the signal energy for the 100 signals in the database and two of the methods compared: SparsEDA and cvxEDA. Blue and green lines show the ratio of SparsEDA and cvxEDA for each particular signal in the database, whereas the horizontal red and black lines display their average values.

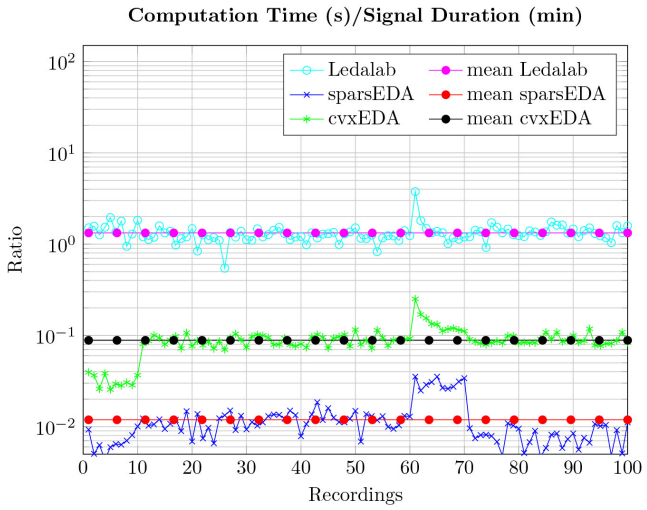

Fig. 5. Ratio between the computation time (in seconds) and the signal's duration (in minutes) for the 100 signals in the database and the three methods compared: SparsEDA, CDA Ledalab and cvxEDA. Cyan, blue and green lines show the ratio of SparsEDA, CDA Ledalab and cvxEDA for each signal in the database, whereas the horizontal purple, red and black lines display their average values.

CDA Ledalab).

- The level of sparsity in the SCR component is directly related to the interpretability of the decomposition obtained: the sparser the signal the more interpretable from a physiological point of view. In this sense, while SparsEDA attains an average sparsity of 0.287% (percentage of non-zero values out of the total number of samples), the other two methods are non-sparse, always returning 100% non-zero values (cvxEDA contains many samples with small but non-null values).
- Confidence intervals and p-values comparing SparsEDA with CDA Ledalab and cvxEDA (both in terms of relative MSE and computation time) have been computed and are displayed in Table II. These values have been computed through the t-test (using Matlab's `ttest2` function) on the hypothesis that the mean of the two distributions of the resuts (i.e., those of SparsEDA and the compared algorithm) are equal. From the p-values obtained (well below the 0.05 threshold typically used) we can see that both the differences in relative computation time and MSE are statistically significant. However, note that the loss in MSE attained by SparsEDA is small with respect to the decrease in computation time, as shown in Figs. 4 and 5 and evidenced by the p-values in Table II.

V. Conclusions

In this paper, we have developed a novel feature extraction method for GSR signals: the SparsEDA algorithm. The main contributions of SparsEDA are the joint estimation of the SCL and SCR components, the multi-scale analysis using an over-complete dictionary, the retrieval of a sparse driver for the SCR component, and the efficient implementation of a fully automated, continuous-mode operation algorithm for on-line processing. The proposed approach has been tested on a database of 100 GSR records from 100 different patients acquired using different sensors and sampling rates, confirming its good performance in terms of relative MSE (-48.50 dB) and computation time (1–2 orders of magnitude lower than other existing algorithms like CDA Ledalab and cvxEDA). Furthermore, the interpretability of the SCR component extracted is enhanced w.r.t. previous approaches, thanks to the sparsity of the driver (unlike both CDA Ledalab and cvxEDA, which return non-sparse drivers) and the lack of artificial non-interpretable signals introduced to minimize the MSE (as done by CDA Ledalab). In summary, SparsEDA confirms the feasibility of developing a fast and fully automated method for extracting the GSR components from large EDA records. One possible future line would be embedding this software within a wearable sensor or a real-time mobile application to detect SNS symptoms using driver activations. Another potential application would be integrating it within medical software to detect stress reactions while an patient is performing some activity.

VI. Acknowledgements

Francisco Hernando-Gallego and Antonio Artés-Rodríguez would like to acknowledge the support obtained from Airbus Defence & Space, specially from SAVIER Open Innovation project members: José Insenser, Roberto Apestigue and Gemma Blasco. David Luengo would like to acknowledge the financial support of the Spanish Ministry of Economy and Competitiveness through project TEC2015-64835-C3-3-R. Antonio Artés would also like to acknowledge the support from MINECO/FEDER (project 'ADVENTURE', id. TEC2015-69868-C2-1-R), and from Comunidad de Madrid (project 'CASI-CAM-CM', id. S2013/ICE-2845). The authors also want to acknowledge Raquel Bailón and Pablo Laguna for providing us with a GSR database to test the SparsEDA method and the Project ES3, that contributed the GSR signal acquisition.

#	Sampling Rate	Sensor	Duration (minutes)		MSE/SE (dB)		Computation Time (seconds) CDA Ledalab		SparsEDA		cvxEDA	
			Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
1-5	4 Hz	Empatica E4	50.00	00.00	-55.90	7.27	78.29	12.36	0.31	0.09	1.65	0.34
6-10	5 Hz	Microsoft Band 2	33.99	19.50	-66.92	19.28	52.51	41.65	0.28	0.23	1.09	0.78
11-60	8 Hz	Medicom MTD	39.43	05.47	-44.03	5.97	48.53	10.37	0.46	0.11	3.50	0.72
61-65	16 Hz	Qsensor	32.62	22.00	-59.88	7.07	41.15	53.46	0.29	0.24	2.73	2.71
66-70	128 Hz	Shimmer 3	9.90	00.00	-53.42	6.62	11.56	1.14	0.14	0.03	1.13	0.03
71-100	8 Hz	Qsensor	9.89	00.00	-48.92	9.20	72.80	10.13	0.41	0.10	4.70	0.55

TABLE I

SUMMARY OF THE CHARACTERISTICS OF THE 100 GSR RECORDS USED IN THE SIMULATIONS AND THE RESULTS OBTAINED. THEY ARE SEPARATED INTO SIX GROUPS ACCORDING TO THE SAMPLING RATE AND THE SENSOR. THE DURATION (MEAN AND STANDARD DEVIATION [STD.]) OF EACH GROUP ARE ALSO SHOWN. RELATIVE MSE (MSE/SIGNAL ENERGY (SE)) IS ONLY SHOWN FOR SPARSEDA. COMPUTATION TIME (MEAN AND STD.) IS SHOWN FOR THE THREE ALGORITHMS COMPARED: SPARSEDA, CDA LEDALAB AND CVXEDA.

#	Sampling Rate	MSE/SE SparsEDA vs cvxEDA			CT/SD SparsEDA vs CDA Ledalab			CT/SD SparsEDA vs cvxEDA		
		p-values	conf. int. lower	conf. int. upper	p-values	conf int. lower	conf int. upper	p-values	conf. int. lower	conf. int. upper
1-5	4 Hz	9.09 e-04	14.09	37.20	6.21 e-07	-1.81	-1.32	2.96 e-05	-0.03	-0.020
6-10	5 Hz	3.40 e-03	16.54	58.84	2.39 e-09	-1.81	-1.05	8.26 e-07	-0.027	-0.019
11-60	8 Hz	8.67 e-06	3.16	7.79	6.77 e-68	-1.25	-1.15	4.01 e-68	-0.080	-0.073
61-65	16 Hz	7.30 e-03	9.35	43.49	3.10 e-10	-2.98	-0.86	2.50 e-04	-0.18	-0.090
66-70	128 Hz	4.81 e-05	22.68	41.41	1.90 e-08	-1.25	-1.02	2.07 e-10	-0.091	-0.081
71-100	8 Hz	6.58 e-07	-28.89	-13.63	-2.21 e-43	-1.43	-1.30	-3.90 e-45	-0.085	-0.077

TABLE II

STATISTICAL COMPARATIVE BETWEEN SPARSEDA AND CVXEDA IN TERMS OF RELATIVE MSE/SE AND SPARSEDA VERSUS CVXEDA AND CDA LEDALAB IN TERMS OF CT/SD. P-VALUES AND CONFIDENCE INTERVALS (LOWER AND UPPER LIMITS) ARE DISPLAYED TO SHOW THE SIGNIFICANT DIFFERENCES. MSE/SE = MSE/SIGNAL ENERGY. CT/SD = COMPUTATION TIME (S)/SIGNAL DURATION (MIN).

REFERENCES

[1] W. Boucsein, *Electrodermal activity*. Springer Science & Business Media, 2012.

[2] S. Taylor, N. Jaques, W. Chen, S. Fedor, A. Sano, and R. Picard, "Automatic Identification of Artifacts in Electrodermal Activity Data," *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 1934–1937, 2015.

[3] V. G. Macefield and B. G. Wallin, "The discharge behaviour of single sympathetic neurones supplying human sweat glands," *Journal of the autonomic nervous system*, vol. 61, no. 3, pp. 277–286, 1996.

[4] B. G. Wallin, "Sympathetic nerve activity underlying electrodermal and cardiovascular reactions in man," *Psychophysiology*, vol. 18, no. 4, pp. 470–476, 1981.

[5] M. M. Bradley and P. J. Lang, "Handbook of psychophysiology," *Handbook of psychophysiology*, 2000.

[6] J. A. Healey and R. W. Picard, "Detecting stress during real-world driving tasks using physiological sensors," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 2, pp. 156–166, 2005.

[7] J. Zhai and A. Barreto, "Stress detection in computer users based on digital signal processing of noninvasive physiological variables," in *Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE*, pp. 1355–1358, IEEE, 2006.

[8] M. Singh and A. Queyam, "Stress Detection in Automobile Drivers using Physiological Parameters: A Review," *International Journal of Electronics Engineering*, no. 2, pp. 1–5, 2013.

[9] J. Hernandez, P. Paredes, A. Roseway, and M. Czerwinski, "Under Pressure: Sensing Stress of Computer Users," *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, pp. 51–60, 2014.

[10] E. Hudlicka, "To feel or not to feel: The role of affect in human–computer interaction," *International journal of human-computer studies*, vol. 59, no. 1, pp. 1–32, 2003.

[11] C. L. Lim, C. Rennie, R. J. Barry, H. Bahramali, I. Lazzaro, B. Manor, and E. Gordon, "Decomposing skin conductance into tonic and phasic components," *International Journal of Psychophysiology*, vol. 25, pp. 97–109, feb 1997.

[12] D. M. Alexander, C. Trengove, P. Johnston, T. Cooper, J. P. August, and E. Gordon, "Separating individual skin conductance responses in a short interstimulus-interval paradigm," *Journal of Neuroscience Methods*, vol. 146, no. 1, pp. 116–123, 2005.

[13] M. Benedek and C. Kaernbach, "Decomposition of skin conductance data by means of nonnegative deconvolution," *Psychophysiology*, vol. 47, no. 4, pp. 647–658, 2010.

[14] M. Benedek and C. Kaernbach, "A continuous measure of phasic electrodermal activity," *Journal of Neuroscience Methods*, vol. 190, no. 1, pp. 80–91, 2010.

[15] A. Greco, G. Valenza, A. Lanata, E. P. Scilingo, and L. Citi, "cvxeda: A convex optimization approach to electrodermal activity processing," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 4, pp. 797–804, 2016.

[16] S. Jain, U. Oswal, K. S. Xu, B. Eriksson, and J. Haupt, "A compressed sensing based decomposition of electrodermal activity signals," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 9, pp. 2142–2151, 2017.

[17] H. Garnier and L. Wang, *Identification of Continuous-time Models from Sampled Data*. Springer Science & Business Media, 2008.

[18] M. Elad, "Sparse and redundant representations: From theory to applications in signal and image processing," 2010.

[19] C. A. Frantzidis, E. Konstantinidis, C. Pappas, and P. D. Bamidis, "An automated system for processing electrodermal activity.," *Studies in health technology and informatics*, vol. 150, p. 787, 2008.

[20] V. G. Macefield, B. G. Wallin, and a. B. Vallbo, "The discharge behaviour of single vasoconstrictor motoneurones in human muscle nerves.," *The Journal of physiology*, vol. 481 Pt 3, pp. 799–809, 1994.

[21] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[22] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *The Annals of statistics - Institute of Mathematical Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[23] S. Monzón, T. Trigano, D. Luengo, and A. Artés-Rodríguez, "Sparse spectral analysis of atrial fibrillation electrograms," in *2012 IEEE*

International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6, 2012.

[24] D. Luengo, S. Monzón, T. Trigano, J. Vía, and A. Artés-Rodríguez, "Blind analysis of atrial fibrillation electrograms: a sparsity-aware formulation," *Integrated Computer-Aided Engineering*, vol. 22, no. 1, pp. 71–85, 2015.

[25] D. Luengo, J. Vía, S. Monzón, T. Trigano, and A. Artés-Rodríguez, "Cross-products LASSO," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6118–6122, 2013.

[26] Q. Wang, "COSBOS: COlor-Sensor-Based Occupancy Sensing." https://se.mathworks.com/matlabcentral/fileexchange/48428-cosbos--color-sensor-based-occupancy-sensing/content/COSBOS_v1.0/LTM_Recovery/Lib/SparseLab2.1-Core/Solvers/SolveLasso.m, 12 Nov. 2014.

[27] J. Hernandez, I. Riobo, A. Rozga, G. D. Abowd, and R. W. Picard, "Using electrodermal activity to recognize ease of engagement in children during social interactions," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 307–317, ACM, 2014.

[28] Ledalab, University of Graz (Austria), "GSR signal example." http://ledalab.de/download/ivn07_16_matlab.mat.

[29] J. Aguiló, P. Ferrer-Salvans, A. García-Rozo, A. Armario, Á. Corbí, F. J. Cambra, R. Bailón, A. González-Marcos, G. Caja, S. Aguiló, *et al.*, "Project ES3: Attempting to quantify and measure the level of stress," *Revista de neurologia*, vol. 61, no. 9, pp. 405–415, 2015.

[30] Medicom MTD Ltd., "Medical equipment for neurophysiology, polysomnography, biofeedback and research." http://www.medicom-mtd.com/en/.

[31] A. Hernando, J. Lázaro, E. Gil, A. Arza, J. M. Garzón, R. López-Antón, C. de la Cámara, P. Laguna, J. Aguiló, and R. Bailón, "Inclusion of respiratory frequency information in heart rate variability analysis for stress assessment," *IEEE journal of biomedical and health informatics*, vol. 20, no. 4, pp. 1016–1025, 2016.

[32] Empatica, "Empatica E4 wristband." https://www.empatica.com/e4-wristband.

[33] Microsoft, "Microsoft band 2." https://www.microsoft.com/microsoft-band/.

[34] Affectiva, "Qsensor." http://qsensor-support.affectiva.com/.

[35] Shimmer Sensing, "Shimmer 3 GSR unit." http://www.shimmersensing.com/products/shimmer3-wireless-gsr-sensor.

Francisco Hernando-Gallego was born in Segovia, Spain, in 1990. He received the B.S. degree in communication systems engineering with honors in 2012, the M.S. degree in telecommunications engineering with honors in 2014 and the M.S. degree in multimedia and communications, all of them from Universidad Carlos III de Madrid, Madrid Spain, where he is currently working toward the Ph.D. degree in machine learning.

His research interests include machine learning applied to several fields, such as, stress recognition, and human behavior characterization and sport betting predictions.

David Luengo (M03) was born in Santander, Spain, in 1974. He received the M.Sc. and Ph.D. degrees in electrical engineering from the Universidad de Cantabria, Spain, in 1998 and 2006, respectively.

From 2003 to 2011, he was an Assistant Professor with the Universidad Carlos III de Madrid, Spain. Since 2011, he has been an Associate Professor at the Universidad Politecnica of Madrid. His research interests include statistical signal processing, Monte Carlo methods, and multitask learning.

Antonio Artés-Rodríguez (IEEE S89, M93, SM01) was born in Alhama de Almera, Spain, in 1963. He received the Ingeniero de Telecomunicacin and Doctor Ingeniero de Telecomunicacin degrees, both from the Universidad Politcnica de Madrid, Spain, in 1988 and 1992, respectively.

He is a Professor at the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Spain. Prior to this he has occupied different teaching positions at Universidad de Vigo, Universidad Politcnica de Madrid, and Universidad de Alcalá, all of them in Spain. He has participated in more than 70 projects and contracts and he has co-authored more that 50 journal papers and more than 100 international conference papers. His research interests include signal processing, machine learning, and information theory methods, and its application to health and the characterization of human behavior.