Circuit Analysis of Matrix-like Resistor Networks for Eliminating Crosstalk in Pressure Sensitive Mats

Carlos Medrano-Sánchez, *Senior Member, IEEE,* Raul Igual-Catalán, *Member, IEEE,* Victor H. Rodríguez-Ontiveros, and Inmaculada Plaza-García, *Senior Member, IEEE,*

Abstract—**Recent advances in piezoresistive materials have opened the possibility of developing pressure sensitive flexible mats that cover large areas. They are composed of arrays of sensitive cells. However, in many configurations proposed in the literature, the measurement of a single cell does not recover the cell resistance itself but the equivalent resistance between the row and column conductive strips that select it. If this effect is not corrected, unloaded regions can appear with non negligible pressure, as a kind of ghost object. Diodes can be placed to overcome the problem, but this makes the fabrication more complex especially for prototypes. In this paper we propose a novel software solution based on a circuit analysis of the mat. The set of cell resistances is obtained from the set of equivalent resistances between row and columns. Several algorithms are compared. For simulated values of the array, the true cell resistance can be recovered with a great accuracy. A good compromise between execution speed and error is achieved by a Newton-Krylov nonlinear solver. Nevertheless, this algorithm presents convergence problems when facing values of a real mat. In this case a fixed-point formulation is more appropriate. For 16x16 arrays, it can achieve relative errors with a mean value of 0.0258 in less than 0.1 s running on a regular PC. The removal of ghost objects is also shown visually.**

Index Terms—**Piezoresistive materials, pressure sensitive mat, circuit analysis, Velostat**

I. Introduction

In recent years the availability of flexible pressure sensor mats has offered new opportunities leading to the development of a wide range of applications in health, artificial intelligence, robotics and games among others. This availability has been boosted by the development of new materials and fabrication processes for flexible pressure sensors [1], [2], [3], [4]. The pressure transduction principles are mainly based on piezoresistivity, piezoelectricity or change of capacitance. In this work we focus on arrays of sensors based on piezoresistivity. The resistance of the material changes when a pressure is applied on it. A detailed model for polymer conductive materials is found in [5]. A linear dependence can be assumed as a reasonable approximation for the conductance versus pressure curve [5], [6].

A typical cell is composed of 2 electrodes and a sensing material. Arrays of sensors can also be built to cover wider surfaces. In many cases, there is a common structure composed of orthogonal conductive lines to form such a kind of arrays. Lines in top and bottom layers are orthogonal and the acquisition system scans them in order to access each cell. For instance in [6] a pressure mat for tele-home care applications was presented (32x32 array covering 39x39 cm^2). The sensor was fabricated by embroidering silver-coated yarns on a light cotton fabric and creating pressure sensitive spots by stamping a conductive polymer at the crossing point of stitches. The acquisition electronics used a transimpedance amplifier, so that the output was proportional to the pressure. In [7] an application for detecting sitting postures was presented (10x10 inch^2, 1/8 inch space between cells). The layer of sensitive material was made of fiber-based yarn coated with piezoelectric polymer. The application of a force led to a squeezing of intrafibers. The buses in the top and bottom layer were orthogonal. They were scanned using analog switches. To read a given spot a row was connected to V_{cc} and a column to an ADC and to ground via an offset resistor. The column voltage was read and the resistance of this voltage divider circuit obtained. In [8] another similar example can be found. An 80x80 matrix was composed of a thin layer of conductive polymer fiber sheet positioned between 80 parallel stripes of conductive foil on each side, spaced 1 cm apart. The mat was used for detecting gym exercises. In [9] a 32x32 sensor matrix was constructed with Velostat material within a top and a bottom electrode layers. The authors of this paper used the sensor for footprint evaluation. The sensor presented in [10] was also based on two conductive layers and a sensitive layer. In the conductive layers, the lines were made by screen printing technology. The pressure sensor layer was a linen cloth embedded with a 32x32 sensor array. The size of the mat was 38.5x36 cm^2, approximately the size of an adult's chest. The mat was used for unobtrusive sleep monitoring. In [11] a device consisting of two composite films was presented. Each film in turn was composed of a polydimethylsiloxane layer, a Cr/Au layer and a patterned conductive film. Each tactile sensing cell included two conductive polymer films with microdomes. The polymer films were fabricated with a variant of the soft-lithography technique. The proposed method achieved high sensitivity and isolation between cell arrays. The technique was demonstrated in an 8x8 array of sensors, with cells separated by 2 mm. On the other hand, in [12] the electrodes were fabricated with an interdigital structure and the conductive polymer was placed on top of them, providing a contact with decreasing resistance as force was applied. A 5x5 flexible force sensor array was shown, covering an area of 84x84 mm^2.

Regardless of the sensor material, if the conductive strips

C. Medrano-Sánchez, R. Igual-Catalán, I. Plaza-García are with EduQTech, E.U. Politecnica , c/ Atarazana 2, Teruel (Spain), and IIS Aragón, University of Zaragoza

V.H. Rodríguez-Ontiveros is with EduQTech, E.U. Politecnica , c/ Atarazana 2, Teruel (Spain), University of Zaragoza

Manuscript received XXXX; revised XXXX.

run along the whole mat, they connect all the cells in a row or column. In this case, an annoying crosstalk effect between spots appears. In a square of four cells, if three of them are pressed the fourth one also appears as slightly pressed when scanning it. This is due to the current flow through the path formed by the three pressed cells [13]. As a consequence, objects appear often as rectangular in shape. In [7] a smoothing algorithm based on Gibbs sampling was proposed. It tended to bring values of suspicious pixels close to the average of the neighbor sites. However, this algorithm only tackled mechanical coupling and not the coupling between separated sites. Cheng et al. [14] used a large mat for activity recognition. A crosstalk effect was acknowledged and a kind of image processing algorithm was proposed. However neither the crosstalk origin nor the details of the algorithm were discussed in length. On the other hand, hardware solutions can also be adopted. Diodes can be placed in series with each cell to avoid the current flow. This is a typical solution in regular keyboards and was also proposed for pressure sensors in [15]. The solution has been adopted in recent designs [16], [9]. The leak currents can also be partially diverted by setting pull-downs resistors in electrodes that are not being addressed. Nevertheless, this does not remove completely the effect. Another option is to set the unused lines to a suitable voltage value, so that the leak currents are completely diverted. In the zero potential solution adopted in [6], [17], [13] the unused lines were set to ground (either directly or as an imaginary zero potential of a transimpedance amplifier). This gives an alternative path for the unwanted currents. However, again this implies the use of more components: one driver or an output amplifier per line. The effect can also be eliminated by building completely isolated cells with their own connections, or at least separated by regions to reduce it. For instance, in the 3x3 array shown in [18] there was a common electrode on one side of the array, and an output line for each sensitive cell on the other side. The same electrode configuration was used in [19] for a 11x13 pixel array. Meyer et al. [20] also configured a single common electrode and then a separated wire for each site. This required 3 different layers to properly route the 240 signals of their prototype. Handling this number of connections is clearly cumbersome for large arrays.

In this paper, we propose a software solution to the crosstalk problem without relying on additional components. It is theoretically well-grounded in the field of circuit analysis. To the best of our knowledge, this is the first time an approach of this kind is proposed for pressure sensitive mats.

The rest of the paper is structured as follows. In section II the problem is stated formally using circuit analysis and several numerical approaches to solve it are presented. In section III we present results from simulated arrays, in which ideal conditions for the algorithms can be tested, and from a home-made mat. Finally, section IV is devoted to conclusions.

II. Theoretical background

A. Problem statement

The pressure sensor array is composed of several cells. Figure 1 shows an example of a 2x2 array. The resistance of the cell intersecting row i and column j is denoted as r_i^j and its conductance as $g_i^j = 1/r_i^j$. To measure one of the cells, typical acquisition systems include multiplexers and demultiplexers that connect a given row and column to some nodes of the measurement circuit. However, since rows and columns cover all the mat, the measured resistance is also influenced by the rest of the network. For instance, figure 1a shows a typical setup in which a simple voltage divider is considered to measure cell $(1, A)$. The current will not only flow through r_1^A but also through the series resistor $r_2^A + r_2^B + r_1^B$. This current flow could be avoided by placing diodes in series with each cell. However, in this paper we are going to consider software solutions that do not require any additional hardware.

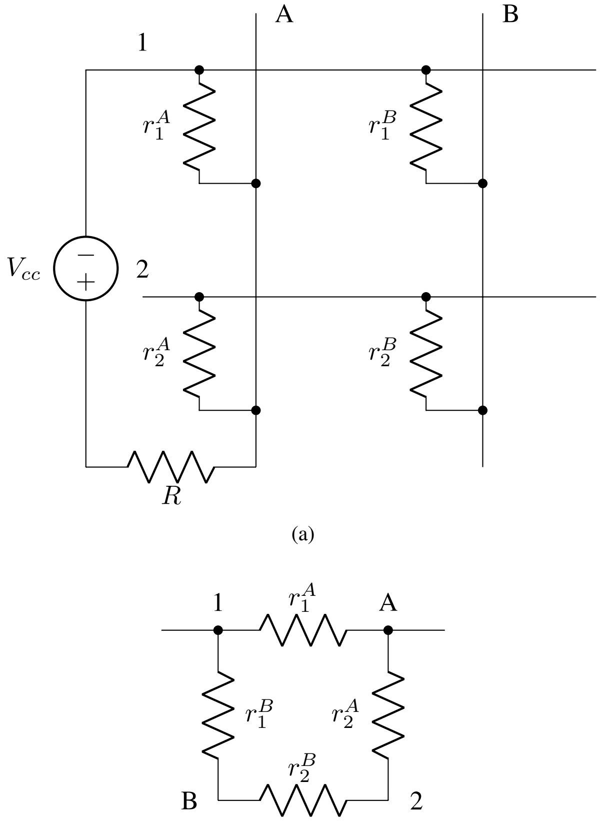


Fig. 1. Example of a 2x2 array. In the top figure the $(1, A)$ cell is connected to a voltage divider to measure it. The bottom figure shows the equivalent resistor seen from nodes $(1, A)$.

Regardless of the signal conditioning circuit used to measure resistance (voltage divider or transimpedance amplifier), the equivalent resistance between row i and column j will be measured if those nodes were connected to it. The equivalent resistance between row i and column j is denoted as R_i^j and its conductance as $G_i^j = 1/R_i^j$. For instance, figure 1b shows the equivalent resistor between the first row and the first column for a 2x2 array. However, it is very difficult, if not impossible, to find easy rules of resistor combination (parallel or series) to obtain the equivalent resistance value except for the 2x2 case. In fact, although the matrix arrangement may suggest that a corner cell is different from a center cell, all of the locations

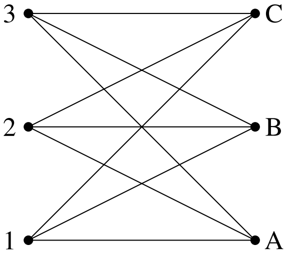


Fig. 2. Schematic representation of the nodes in a 3x3 array. The links have a resistance value (not shown for simplicity).

are equivalent since each row is connected to all the columns and vice versa. Figure 2 represents a 3x3 array to highlight this point. Each link has an associated resistance not shown for simplicity.

Thus, the equivalent resistances R_i^j (or conductances G_i^j) are measured by scanning the rows and columns in the data acquisition system. The problem is to find the cell resistances r_i^j (or conductances g_i^j) from them, so that the pressure in each cell can be determined.

B. Circuit Analysis of the Resistor Array

A standard circuit analysis of the network is applied in this section to find the relation between the set of conductances G_i^j and the set of conductanges g_i^j. For the sake of simplicity, first a 3x3 network is considered as an example, figure 3. Formally, the equivalent resistance between row 2 and column C can be obtained by solving the circuit of the figure, in which a current source is placed between colum C and row 2. If V_i denotes the voltage at row i and V^j the voltage at column j, the equivalent conductance is:

$$G_2^C = \frac{1}{R_2^C} = \frac{I_{ref}}{V^C - V_2} \tag{1}$$

I_{ref} is just a multiplicative constant in the solution that can be set to one in the implementation.

Without loss of generality, row 1 can be taken as reference so that $V_1 = 0$. The circuit can be solved by applying Kirchoff's law to the rest of the nodes. The solution is found by solving the following linear system:

$$\mathbf{CV} = \mathbf{I} \tag{2}$$

where $\mathbf{V}$ is the vector of unknown voltages $\mathbf{V} = (V_2, V_3, V^A, V^B, V^C)^{\mathrm{T}}$, $\mathbf{I} = (-I_{ref}, 0, 0, 0, I_{ref})^{\mathrm{T}}$ and $\mathbf{C}$ is the conductance matrix of the linear system of equations:

$$\mathbf{C} = \begin{pmatrix} g_2 & 0 & -g_2^A & -g_2^B & -g_2^C \\ 0 & g_3 & -g_3^A & -g_3^B & -g_3^C \\ -g_2^A & -g_3^A & g^A & 0 & 0 \\ -g_2^B & -g_3^B & 0 & g^B & 0 \\ -g_2^C & -g_3^C & 0 & 0 & g^C \end{pmatrix}$$

where the shorthand notations $g_i = \sum_j g_i^j$ and $g^j = \sum_i g_i^j$ have been used. For instance: $g_3 = g_3^A + g_3^B + g_3^C$.

After solving equation 2, the equivalent resistance and conductance are obtained from equation 1.

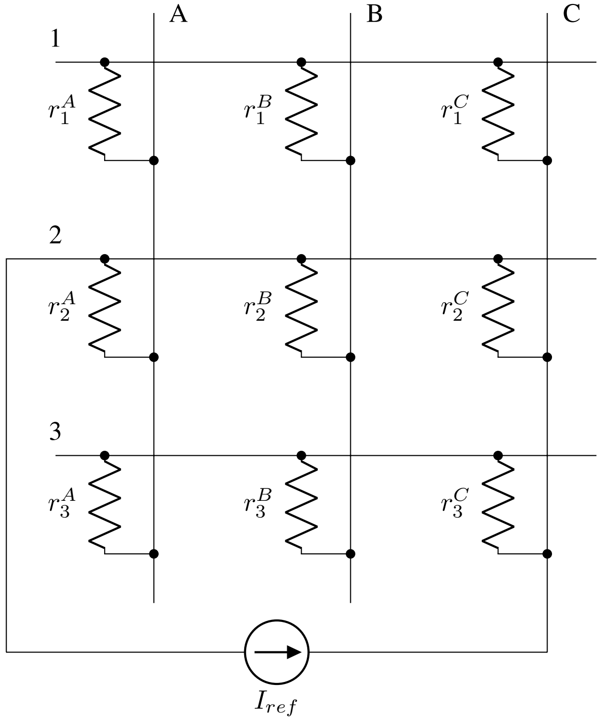


Fig. 3. Circuit to determine the equivalent resistance between row 2 and column C in a 3x3 array.

To find the equivalent resistance of any other pair of row-column nodes, the only vector that has to be changed in equation 2 is $\mathbf{I}$. For instance, to find R_3^A, the intensity vector $\mathbf{I} = (0, -I_{ref}, I_{ref}, 0, 0)^T$ has to be used, while to find R_1^C the intensity vector is $\mathbf{I} = (0, 0, 0, 0, I_{ref})^T$. Note that in the latter case there is no $-I_{ref}$ term in the intensity vector since the node voltage V_1 is taken as ground so that it does not appear in the unknown vector $\mathbf{V}$.

Let us generalize for an NxM resistor array. Assuming that $V_1 = 0$ (ground), there will be N+M-1 equations. To find the equivalent resistance between row i and column j, the following equations must be solved to find node voltages:

$$\sum_{q=1}^{M} g_p^q (V_p - V^q) = -I_{ref}\delta_{pi}, \; p = 2, \ldots, N \tag{3}$$

$$g_1^q V^q + \sum_{p=2}^{N} g_p^q (V^q - V_p) = I_{ref}\delta_{qj}, \; q = 1, \cdots, M \tag{4}$$

where δ_{kl} is the Kronecker delta (1 if $k = l$, 0 otherwise).

Then the equivalent conductance is found as:

$$G_i^j = \frac{I_{ref}}{V^j - V_i} \tag{5}$$

These equations allow finding numerically the relation between the cell conductances and the equivalent conductances between the nodes. If $\mathbf{G}$ is the matrix of equivalent conductances G_i^j and $\mathbf{g}$ the matrix of all cell conductances g_i^j, the theoretical relation $\mathbf{G} = \mathbf{F}(\mathbf{g})$ can be found numerically using the algorithm shown in figure 4. For each component of the

```
Input: matrix of cell conductances $g_i^j, i = 1,\cdots,N\ j = 1,\cdots,M$
Output: matrix of equivalent conductances $G_i^j, i = 1,\cdots,N\ j = 1,\cdots,M$
for all pairs $(i,j)$ do
    Solve equations 3 and 4
    $G_i^j \leftarrow \frac{I_{ref}}{V^j - V_i}$
end for
```

Fig. 4. Algorithm to find equivalent conductance matrix $\mathbf{G} = \mathbf{F}(\mathbf{g})$.

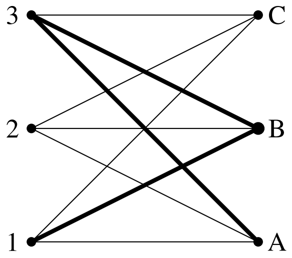


Fig. 5. Schematic representation the lowest conductance path between nodes 1 and A.

equivalent conductance matrix, the notation $G_i^j = F_i^j(\mathbf{g})$ will be used in this paper.

C. Proposed numerical solutions

The problem is finding the matrix $\mathbf{g}$ from the experimental values of the equivalent conductances $\mathbf{G}$, that is, to invert the function $\mathbf{G} = \mathbf{F}(\mathbf{g})$ defined in figure 4. In this section several approaches are proposed. The algorithms were implemented in Python, relying on the scipy package (version 0.19.1) [21] whenever possible. The algorithms were run on a regular PC running Ubuntu 18.04 (Intel(R) Core i5-4570 CPU, 3.2 GHz, 4G RAM).

1) Heuristic approach: The heuristic approach is a crude approximation. From the multiple paths that the current can flow between row i and column j, only the intersection cell and the lowest resistance path are taken into account. To find the lowest resistance path, the approximation $g_p^q \approx G_p^q$ is considered. For instance, in figure 5 let us assume that the lowest resistance path between row 1 and column A is the path highlighted by the thicker links: row 1 to column B, column B to row 3 and row 3 to column A. Then, it is assumed that the resistance between row 1 and column A can be calculated as the parallel of the cell resistance r_1^A and the lowest path resistance $r_1^B + r_3^B + r_3^A$. Therefore, in terms of conductances:

$$G_1^A \approx g_1^A + \left(\frac{1}{g_1^B} + \frac{1}{g_3^B} + \frac{1}{g_3^A}\right)^{-1} \tag{6}$$

Then, a further assumption is taken in the right hand side: $g_1^B \approx G_1^B$, $g_3^B \approx G_3^B$, $g_3^A \approx G_3^A$. Thus, the cell conductance can be found from equation 6:

$$g_1^A = G_1^A - \left(\frac{1}{G_1^B} + \frac{1}{G_3^B} + \frac{1}{G_3^A}\right)^{-1} \tag{7}$$

Thus, the heuristic approximation is a very fast single step correction. It also ensures that $g_i^j < G_i^j$, which is true in general. However, it is not based on a formal approximation to the problem and the assumptions are not justified in general.

2) Least square solution: The problem can also be stated as a least square problem. The elements of $\mathbf{g}$ are found by minimizing the following cost function:

$$\|\mathbf{G} - \mathbf{F}(\mathbf{g})\|^2 = \sum_{i=1}^{N}\sum_{j=1}^{M} \|G_i^j - F_i^j(\mathbf{g})\|^2 \tag{8}$$

The cell conductances are also subjected to an additional constraint:

$$0 \leq g_i^j \tag{9}$$

To implement this approach, the functions scipy.optimize.leastsq and scipy.optimize.least_squares [21] were used. The first one is based on a Levenberg-Marquardt algorithm. It does not handle the constraints but in some cases they are not required as shown in the experimental section. The second one can handle constraints and uses a Trust Region Reflective algorithm [21].

3) Nonlinear equation solver: Newton-Krylov: The problem can also be stated as the solution of the nonlinear equation:

$$\mathbf{G} - \mathbf{F}(\mathbf{g}) = 0 \tag{10}$$

Among the solvers offered by the scipy package, the one that did converge in a reasonable amount of time was the newton_krylov solver of the package scipy.optimize (more details about convergence problems are presented in the experimental section). According to the documentation of the package, these methods can deal with large nonlinear problems. This is because the kind of methods implemented by this function require only evaluating the Jacobian-vector products, which are conveniently approximated by a finite difference [22].

4) Nonlinear equation solver: fixed-point formulation: The nonlinear equation can also be stated as a fixed-point formulation. Generally speaking, fixed-points equations are in the form $\mathbf{x} = \mathbf{H}(\mathbf{x})$, where $\mathbf{x}$ is a vector. The Picard iterations can be used to solve the problem, finding the next root approximation, $\mathbf{x}_{n+1}$, from the previous one $\mathbf{x}_n$ [22]:

$$\mathbf{x}_{n+1} = \mathbf{H}(\mathbf{x}_n) \tag{11}$$

To secure convergence, the Picard iteration must be damped in many cases by introducing the β parameter [22]:

$$\mathbf{x}_{n+1} = (1-\beta)\mathbf{x}_n + \beta\mathbf{H}(\mathbf{x}_n) \tag{12}$$

Returning to our problem, it can also be formulated in this way. For that purpose the equation to be solved is first set as $0 = \mathbf{G} - \mathbf{F}(\mathbf{g})$. Then, adding $\mathbf{g}$ to both the left and right sides leads to:

$$\mathbf{g} = \mathbf{g} + \mathbf{G} - \mathbf{F}(\mathbf{g}) \tag{13}$$

which is the standard form to apply a solution based on Picard iterations as follows:

```
Input: $\mathbf{G}$,$\beta$, $f_{tol}$ and $n_{max}$ and initial iterate ($\mathbf{g}_0$)
Output: Approximated solution $\mathbf{g}$
$n \leftarrow 0$
$residual \leftarrow \max(|G_i^j - F_i^j(\mathbf{g}_0)|)$
while $n < n_{max}$ and $residual > f_{tol}$ do
   $\mathbf{g}_{n+1} \leftarrow \mathbf{g}_n + \beta(\mathbf{G} - \mathbf{F}(\mathbf{g}_n))$
   for all pairs $(i, j)$ do
      if $(g_i^j)_{n+1} < 0$ then
         $(g_i^j)_{n+1} \leftarrow 0$
      end if
   end for
   $residual \leftarrow \max(|G_i^j - F_i^j(\mathbf{g}_{n+1})|)$
   $n \leftarrow n + 1$
end while
return $\mathbf{g}_n$
```

Fig. 6. Algorithm for fixed-point solution.

$$\mathbf{g}_{n+1} = \mathbf{g}_n + \mathbf{G} - \mathbf{F}(\mathbf{g}_n) \tag{14}$$

A dumping factor β is introduced leading to the final form:

$$\begin{aligned}\mathbf{g}_{n+1} &= (1-\beta)\mathbf{g}_n + \beta(\mathbf{g}_n + \mathbf{G} - \mathbf{F}(\mathbf{g}_n)) \\ &= \mathbf{g}_n + \beta(\mathbf{G} - \mathbf{F}(\mathbf{g}_n))\end{aligned} \tag{15}$$

The algorithm can be easily implemented. Two further conditions are added. One is the termination condition, which we have borrowed from the scipy function newton_krylov, considering a tolerance parameter f_{tol} and a maximum number of iterations n_{max}, so that the algorithm is terminated if the iterations achieve n_{max} or if the max norm of $\mathbf{G} - \mathbf{F}(\mathbf{g})$ is less than f_{tol}. Besides, the positiveness of the conductances is reinforced in each iteration by bringing to zero those conductances in which the damped Picard iteration gives a negative value. The algorithm is shown in figure 6

III. Results and discussion

To verify and compare the proposed approaches, values obtained from simulations or from a real pressure sensitive mat were used.

A. Simulated arrays of resistors

In this section the results from the simulations are presented. The key idea was to generate random arrays of 16x16 cell conductances, $\mathbf{g}$. From them, the equivalent conductance matrix $\mathbf{G}$ was found using the algorithm in figure 4. Then, the cell conductances were recovered from $\mathbf{G}$ using the approaches proposed in section II-C and compared with the original ones. These simulations constitute a first test for the approaches in ideal conditions (there were no errors other than the numerical ones). Besides the solution was known and could be compared directly with the output of the solvers.

For these simulated arrays, the positiveness of the conductance solution was not imposed since the solution already gave positive values except for few rare cases.

TABLE I

Performance of the algorithms with random arrays of conductances. Mean and standard deviation (in parentheses) of MARE and execution time are given from 100 simulations. The 95% percentile is obtained from all the cell errors (100×256 values) in the total set of simulations.

Algorithm	Time (s)	MARE	95% percentile
No correction	-	24.3(6.1)	71.3
Heuristic	$6.41(0.36)e{-}02$	13.6(3.4)	40.2
Least square (default)	5.45(0.21)	$8.3(3.9)e-15$	$2.4e-14$
Least square	3.27(0.59)	$3.0(12)e-05$	$3.9e-05$
Newton-Krylov (default)	0.208(0.019)	$4.2(4.1)e-07$	$1.1e-06$
Newton-Krylov	0.192(0.049)	$2.2(4.9)e-04$	$5.8e-04$
Fixed-Point (default)	2.00(0.16)	$2.1(1.1)e-10$	$5.7e-10$
Fixed-Point	0.747(0.088)	$1.54(0.59)e{-}04$	$4.9e-04$

If the solution found is denoted by $(g_i^j)'$, the absolute relative error (ARE) for a cell was:

$$\epsilon_i^j = \left|\frac{(g_i^j)' - g_i^j}{g_i^j}\right| \tag{16}$$

The mean absolute relative error (MARE) for each array and the execution time were obtained as the main figures of merit. Each simulation was repeated 100 times to gather statistics. Thus, 100 values of MARE and execution time were obtained from which the mean and the standard deviation are going to be shown. Besides, all the cell errors during the whole run (100×256 values) were also stored from which the 95% percentile was extracted.

All algorithms except the heuristic approach have some internal parameters that can be tuned to get a trade-off between execution time and error. Thus, they were tested twice. Firstly the default parameters provided by the numerical package (or similar conditions for fixed-point solution) were selected. Secondly, the parameters were tuned to have a 95% percentile of ARE less than 0.001. For instance, this value is close to the resolution of a 10 bit ADC, which seems a reasonable threshold.

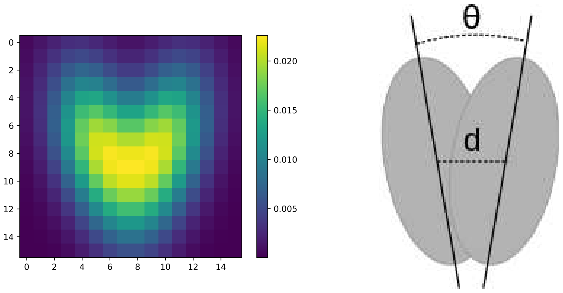
Two kinds of simulations were performed:

- In the first one, conductances were just random numbers between a minimum and a maximum value with no relation between neighbor cells.
- In the second one, a kind of object composed of two support zones was simulated (mimicking feet or a pressure map from a seated person)

Each simulation is described in a separate subsection.

1) Uncorrelated random conductance arrays: In this case each cell conductance was a random number following a uniform distribution between $g_{min} = 2e{-}05$ and $g_{max} = 2e{-}02$ with no correlation between neighbor sites. These bounds are typical values in the applications we are interested in. The results are shown in table I, in which the lack of correction is also included for comparison, that is, taking $g_i^j \approx G_i^j$.

It it clear from table I that the lack of correction is not an option if one wants to have a reasonable approximation to the conductance, and therefore to the pressure exerted on the



(a) (b)

Fig. 7. The left figure is an example of a generated conductance map (in S). The right figure is a schematic representation of the distribution showing the two parameters that have a random component, d and θ.

mat. The heuristic approach reduces a lot the MARE (from 24 to 13) but it is still high. Among the formal approaches, these results confirm their soundness since error is almost zero for practical purposes. The execution time ranges from a few seconds to 0.192 s. If a single array is to be corrected all of them could be acceptable. However for processing in real time a sequence of arrays the Newton-Krylov method would be preferred. As expected, processing time can be lowered at the expense of increasing the error. Anyway, it is reasonable (less than 0.001 for 95% percentile)

2) Simulating pressure maps with two regions: In this case, conductances at each position of the matrix, $g(\mathbf{r})$ with $\mathbf{r} = (x, y)^T$, were generated using these equations:

$$g_1(\mathbf{r}) = (g_{max} - g_{min})(\exp(-\frac{1}{2}(\mathbf{r}-\mathbf{r}_1)^T \mathbf{\Sigma_1}(\mathbf{r}-\mathbf{r}_1)))$$
$$g_2(\mathbf{r}) = (g_{max} - g_{min})(\exp(-\frac{1}{2}(\mathbf{r}-\mathbf{r}_2)^T \mathbf{\Sigma_2}(\mathbf{r}-\mathbf{r}_2)))$$
$$g(\mathbf{r}) = g_{min} + (g_1(\mathbf{r}) + g_2(\mathbf{r})) \quad (17)$$

The rationale behind these equations is the following: Since in many pressure sensitive mats conductance is proportional to pressure as a reasonable approximation, the conductance map is also a representation of the pressure map. That is, the pressure is distributed in two regions of centers $\mathbf{r}_1$, $\mathbf{r}_2$ with a 2D Gaussian-like dependence. An example is shown in figure 7. This has some similarity with the pressure distributions that can be found when standing or sitting on the mat, in which the weight is supported by body parts (feet or ischial tuberosities).

To generate several pressure distributions, the separation and the angle between the regions had a random component, see figure 7b.

The results for this kind of pressure shapes are found in table II.

The lack of correction and the heuristic approach give again bad error values. The formal approaches lead to errors higher than in the case of uncorrelated conductances, table I, but the limit of 0.001 for the 95% percentile is still feasible. Computation time is also worse than in table I, especially with the fixed-point approach that increases by more than an order of magnitude. It seems that this kind of conductance maps, closer to real measurements, is hard to be solved by the formal methods.

TABLE II
PERFORMANCE OF THE ALGORITHMS WITH RANDOM ARRAYS OF TWO-REGION CONDUCTANCES. MEAN AND STANDARD DEVIATION (IN PARENTHESES) OF MARE AND EXECUTION TIME ARE GIVEN FROM 100 SIMULATIONS. THE 95% PERCENTILE IS OBTAINED FROM ALL THE CELL ERRORS (100×256 VALUES) IN THE TOTAL SET OF SIMULATIONS.

Algorithm	Time (s)	MARE	95% percentile
No correction	-	11.6(1.6)	33.8
Heuristic	$6.42(0.18)e{-}02$	2.36(0.59)	5.24
Least square (default)	7.1(1.3)	$1.2(2.7)e-10$	$1.6e-11$
Least square	4.31(0.52)	0.018(0.12)	$4.2e-04$
Newton-Krylov (default)	0.402(0.098)	$1.1(1.6)e-03$	0.0026
Newton-Krylov	0.54(0.13)	$5.2(10)e-05$	$1.2e-04$
Fixed-Point (default)	31(10)	$1.3(3.3)e-05$	$1.2e-06$
Fixed-Point	13.6(5.2)	$2.3(3.3)e-04$	$4.5e-04$

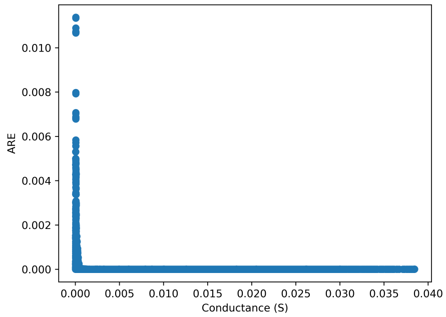


Fig. 8. Values of ARE for all the simulations as a function of conductance for the Newton-Krylov approach.

To check whether the relative error depends on conductance, figure 8 shows a plot of ARE versus conductance for the Newton-Krylov approach (non default parameter values). Each point corresponds to a cell and the whole run of 100 arrays is represented. The error is higher for low values of conductances. This trend is observed for all the formal approaches. The presence of some points with higher error at low conductances also explains why the mean MARE error is higher than the 95% percentile in some rows of table II.

B. Experiments with a real pressure sensitive mat

In this subsection the results of the algorithms tested on measurements of a real mat are presented. We fabricated a pressure sensitive mat composed of three layers. The two outer layers were made of a flexible grid printed with a 3D printer, in which conductive lines were formed on one of the sides using copper adhesive tape 7 mm width. The layers were set with the lines orthogonal. Then, a piezoelectric sheet of Velostat was placed in the middle [23]. This material has become very popular for low-cost applications of pressure sensors [9], [24], [25], [26], [27]. In total, a 16x16 sensor array was built (32x32 cm^2). The copper rows and columns are continuous along the length of the mat, so that the conditions considered in this paper apply. The sensitive cells were formed at the crossing points of row and columns.

TABLE III
PERFORMANCE OF THE ALGORITHMS WITH EXPERIMENTAL VALUES OF CONDUCTANCES. MEAN AND STANDARD DEVIATION (IN PARENTHESES) OF MARE AND EXECUTION TIME ARE GIVEN FROM 360 PRESSURE MAPS OBTAINED. THE 95% PERCENTILE IS OBTAINED FROM ALL THE CELL ERRORS (360×256 VALUES) IN THE TOTAL SET OF ACQUIRED MAPS.

Algorithm	Time (s)	MARE	95% percentile
No correction	-	8.58(0.07)	13.06
Heuristic	$6.53(0.35)e{-}02$	0.85(0.59)	1.61
Least square (default)	51(10)	$1.77(0.47)e{-}02$	0.079
Least square	8.26(0.29)	$5.9(1.8)e-02$	0.19
Newton-Krylov	0.15(0.31)	1.5(2.8)	9.5
Fixed-Point	$9.73(0.67)e{-}02$	$2.58(0.50)e{-}02$	0.092

The measurement circuit consisted mainly of a communication unit, two analog multiplexers and a microcontroller. The microcontroller activated sequentially rows and columns so that the column was connected through a resistor to V_{cc} and the row to ground. Thus a voltage divider circuit was formed, similar to the one shown in figure 1a to measure cell $(1, A)$. The voltage at the column was measured with the ADC of the microcontroller. The whole mat could be sampled each $0.1\,s$ (10 Hz). The raw measurement was sent via Bluetooth to a PC, in which the data were analyzed in an off-line mode.

Unlike in the case of simulations, the true value of each cell conductance was not known for a real mat. Therefore, the error in this case was measured with respect to the equivalent conductance obtained at each cell, taking as the true value the measured one, $\mathbf{G}$. Thus, if $\mathbf{g}'$ was the solution obtained by one of the algorithms, then the equivalent conductance was obtained as $\mathbf{G}' = \mathbf{F}(\mathbf{g}')$, and the absolute relative error at each cell was defined as:

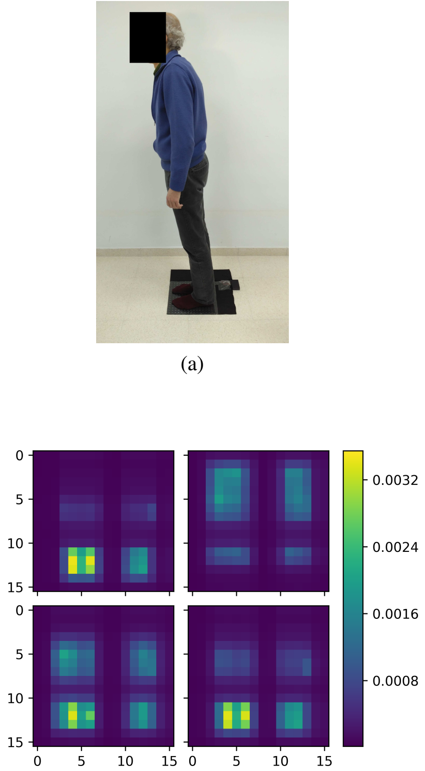
$$\epsilon_i^j = \left| \frac{(G_i^j)' - G_i^j}{G_i^j} \right| \tag{18}$$

Apart from the error definition, the figures of merit were the same as in the case of simulated conductances. For each array the MARE and the execution time were obtained. A set of several conductance maps were acquired to gather statistics. Besides, the 95 % percentile from all the cell errors was also computed.

1) Standing position: independent arrays: In the first experiment with real mats, a subject was standing on the mat in a comfortable position. He was asked to bend slightly the body and then move in circles. A total of 360 arrays were collected. In this subsection, each acquired array is taken independently. Figure 9 shows and image of the experiment and 4 typical conductance maps obtained.

The summary of performance is shown in table III. Unlike in the case of simulations, the constraint of positive conductance was applied when possible (Least Square and fixed-point options). If not, many cell conductances converged to negative values. The Newton-Krylov with default parameters is not shown since the associated package function often got stuck and gave exception due to lack of convergence.

Several aspects are worth highlighting from these results and the conditions under which they were obtained. The errors are higher than in the simulations. Thus, the parameters of the algorithms could not be tuned to reach a 0.001 error for the value of the 95% percentile. This turned out to be too fine for real experiments. Therefore, only for the Least Square approach the results for two sets of parameters are presented in table III. The default configuration is too slow, almost 1 min for processing each array (51 s). The computation time can be reduced by changing the parameters but it is not competitive for real-time applications.


[redacted]

(a)

(b)

Fig. 9. Images of the experiment: The subject an a typical position (top) and four values of conductance maps (bottom). Conductance is in S.

The Newton-Krylov approach, which seemed to be the most suitable for simulated conductances, gave unacceptable errors when the values were obtained from real mats. Its parameters had to be tuned to avoid convergence problems but there was no way to reduce the errors further. Besides, this algorithm does not take into account constraints and many conductance solutions had negative values.

Concerning the fixed-point approach, 25 iterations were used since the error almost did not decrease beyond this number. Thus, only a set of parameters is shown in table III, which already led to the best execution time, 0.0973 s. This value could allow a real-time analysis in this particular case since the whole array was sampled at 10 Hz. The 95% percentile error was not far from that of the Least Square approach with default parameters.

To sum up, in this case the fixed-point solution could be selected since it has several advantages over the other approaches. It is likely that intrinsic errors in the data acquisition system have an influence on the error of the mathematical

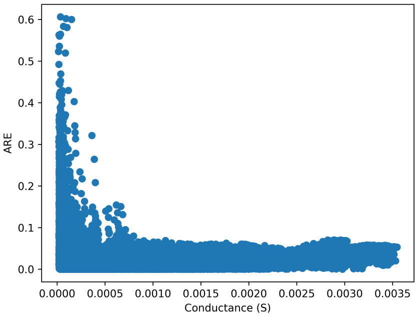


Fig. 10. Values of ARE for all the experimental conductance maps using the fixed-point solution.

solution. In figure 10 the errors as a function of the conductance are shown. The higher errors are concentrated in the low conductance part of the graph. This can be due to numerical aspects of the algorithms, since the simulations shows the same trend, figure 8, or to measurement errors.

The inconsistency between the results of the Newton-Krylov approach with simulated and experimental values deserves an explanation. One of the reasons can be the presence of measurement errors. As a matter of fact, we tested that using simulated values with an injection of random noise as low as 4% in the equivalent conductance matrix $\mathbf{G}$ enabled us to reproduce the problems found with the real mat: many cell conductances turned out to be negative and convergence problems arose. On the other hand, the fixed-point approach, although less sophisticated, allows a modification to reinforce positiveness. Furthermore, it never gets stuck and can be easily tuned with the β parameter to obtain smooth corrections at each time step and with the number of iterations to control execution time.

2) Standing position: sequential measures: The set of conductance maps is the same as in the previous subsection but two initializations were tested. With a single measurement of the matrix, the initial iterate $\mathbf{g}_0 = \mathbf{G}$ was taken in previous sections with a maximum number of 25 iterations in the fixed-point formulation. However, for a sequence of measurements it is reasonable to expect that the solution at time t can be used as the initial iterate for $t+1$ since the difference between two consecutive pressure distributions should not be very high. In this way we wanted to take advantage of the sequential nature of the data using the fixed-point formulation and reducing the maximum number of iterations to 15. Figure 11 shows the execution time and the error as a function of time. In this case, taking a better initial iterate decreased the execution time without changing the error: In the bottom graph, the lines corresponding to the two initializations are almost indistinguishable. In fact, the average error with the new tested initialization is 0.0258, exactly as in table III.

Therefore, in a sequence of real pressure maps a further improvement in processing time can be achieved by selecting a suitable initial iterate. This does not seem to influence the error.

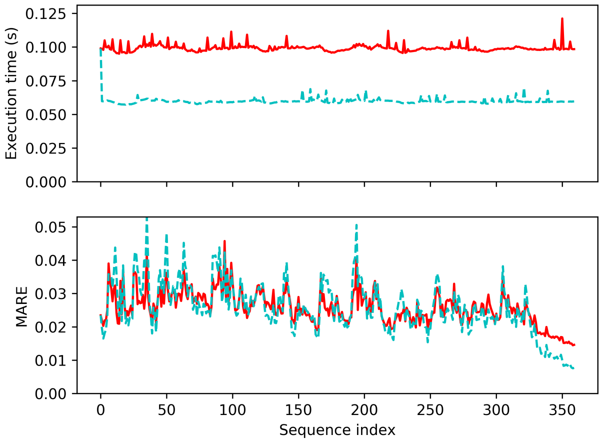


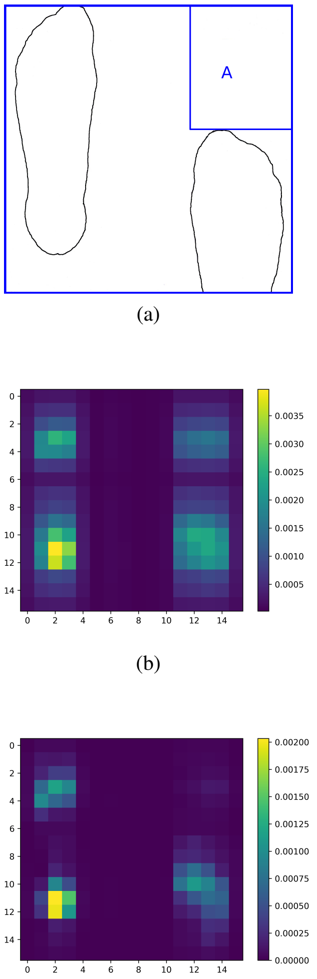
Fig. 11. Time execution and MARE for a sequence of conductance maps. The solution of a map feeds the first iterate of the next one (blue dashed line) or the initial iterate is $\mathbf{G}$ (red solid line).

TABLE IV
PERCENTAGE OF WEIGHT WITHIN REGION A IN FIGURE 12A BEFORE AND AFTER CORRECTION.

No correction	Heuristic	Least-Square	Fixed-point
16.3%	10.6%	3.5%	4.6%

3) Removal of ghost object: The effect of crosstalk in pressure sensitive maps is plain if the pressure is represented as an image. There can be ghost objects that correspond to zones with fictitious pressure where there is no real object on. In this subsection, an experiment to highlight the removal of a ghost object is presented. A subject stood on the mat with a relative displacement between the feet along the anterio-posterior direction. Figure 12 shows a scheme and the results of the experiment. Figure 12a presents the relative position of the feet and an outline of the external feet shape. Region A, in the upper right corner, is empty. Figure 12b shows the conductance map obtained directly from the measurement (we assume that pressure and conductance are proportional, which is reasonable for Velostat). The support zones appeared with a rather rectangular shape. Besides, there was some fictitious pressure in region A. It would be hard to deduce the relative displacement of the feet from that image. After the correction, the ghost object almost disappeared (figure 12c) and the foot shape was not so rectangular.

A quantitative measure of the ghost object removal was obtained by calculating the percentage of weight in region A before and after correction. This is shown in table IV. The ideal value is 0. Without correction, the ghost object seems to carry a 16.3% of the weight. The corrections lower this value. The value obtained with the Least-Square approach is the closest to the ideal although not very different from the fixed-point approach. The correction reduces the fictitious weight by a factor up to 4.7.



(a)

(b)

(c)

Fig. 12. Visual effect of removing ghost objects in area A. Conductance maps are measured in S. Schematic representation of feet (a), measured conductance map (b) and corrected conductance map (c).

IV. Conclusions

Pressure sensitive mats made of resistive sensor arrays present a crosstalk problem if the conductive lines run all along the mat. In this case, ghost objects can appear in zones without pressure exerted on them. In this paper, we have presented several algorithms to remove this effect after analyzing the array with circuit theory. The key point is that the value returned by the measurement system when a cell is scanned is the equivalent resistance between a given row-column pair, instead of the desired cell resistance. For simulated values of resistor networks, several algorithms can recover the true value of the array with negligible error. The Newton-Krylov is the fastest method among them. On the other hand, for experiments with a real mat, the error is higher and a fixed-point approach is best suited. It can be easily tuned to find a compromise between error and computation time. It can find a solution in less than 0.1 s for 16x16 arrays. We have also shown visually that a ghost object disappears after the application of the algorithm. The Newton-Krylov method seems to be very sensitive to measurement errors.

The software solutions can be easily implemented in any kind of prototype or commercial product since they rely on standard numerical packages. Furthermore, their existence has some implications for manufacturing processes. The advantage is clear especially for home made prototypes. The proposed approaches avoid the annoying addition and manual soldering of components (for instance, one diode per cell). For standard production processes the integration of components in flexible printed circuit boards would not be so hard, yet it leads to a higher cost and possibility of component failure. With respect to new research developments on sensor fabrication, the integration with reliable electric contacts and connectors is still a challenge, as pointed out in [4]. Keeping the connections to a minimum is valuable in this context and the proposed methods allow the typical row-column addressing setup, instead of manufacturing a separated connection for each cell.

The solutions presented in this paper can be easily extended to arrays of capacitive sensors or to include the pull-up or pull-down resistors in conductive lines that many designs incorporate.

There are some error sources that are not covered by the proposed approach. For instance, there is also a crosstalk between neighbor cells due to mechanical coupling. Our solutions does not deal with this kind of interference.

Acknowledgment

The authors would like to thank the ”Fondo Social Europeo (Programa Operativo FEDER 2014-2020 Construyendo Europa desde Aragón) and the ”Diputación General de Aragón” (T49_17R) for their financial support. Victor H. Rodriguez acknowledges a grant from CONACYT-Gobierno del Estado de Durango, México 330795/386043.

References

[1] F. Xu, X. Li, Y. Shi, L. Li, W. Wang, L. He, and R. Liu, “Recent developments for flexible pressure sensors: A review,” *Micromachines*, vol. 9, no. 11, p. 580, Nov 2018. [Online]. Available: http://dx.doi.org/10.3390/mi9110580

[2] J. Li, R. Bao, J. Tao, Y. Peng, and C. Pan, “Recent progress in flexible pressure sensor arrays: from design to applications,” *J. Mater. Chem. C*, vol. 6, pp. 11 878–11 892, 2018. [Online]. Available: http://dx.doi.org/10.1039/C8TC02946F

[3] Y. Zang, F. Zhang, C.-a. Di, and D. Zhu, “Advances of flexible pressure sensors toward artificial intelligence and health care applications,” *Mater. Horiz.*, vol. 2, pp. 140–156, 2015. [Online]. Available: http://dx.doi.org/10.1039/C4MH00147H

[4] Y. Ding, T. Xu, O. Onyilagha, H. Fong, and Z. Zhu, “Recent advances in flexible and wearable pressure sensors based on piezoresistive 3d monolithic conductive sponges,” *ACS Applied Materials & Interfaces*, vol. 11, no. 7, pp. 6685–6704, 2019. [Online]. Available: https://doi.org/10.1021/acsami.8b20929

[5] M. Kalantari, J. Dargahi, J. Kvecses, M. G. Mardasi, and S. Nouri, “A new approach for modeling piezoresistive force sensors based on semiconductive polymer composites,” *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 3, pp. 572–581, June 2012.

[6] J. F. Saenz-Cogollo, M. Pau, B. Fraboni, and A. Bonfiglio, “Pressure mapping mat for tele-home care applications,” *Sensors*, vol. 16, no. 3, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/3/365

[7] W. Xu, M. Huang, N. Amini, L. He, and M. Sarrafzadeh, “ecushion: A textile pressure sensor array design and calibration for sitting posture analysis,” *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3926–3934, Oct 2013.

[8] M. Sundholm, J. Cheng, B. Zhou, A. Sethi, and P. Lukowicz, "Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '14. New York, NY, USA: ACM, 2014, pp. 373–382. [Online]. Available: http://doi.acm.org/10.1145/2632048.2636088

[9] S. S. Suprapto, A. W. Setiawan, H. Zakaria, W. Adiprawita, and B. Supartono, "Low-cost pressure sensor matrix using velostat," in *2017 5th International Conference on Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME)*, Nov 2017, pp. 137–140.

[10] W. Li, C. Sun, W. Yuan, W. Gu, Z. Cui, and W. Chen, "Smart mat system with pressure sensor array for unobtrusive sleep monitoring," in *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, July 2017, pp. 177–180.

[11] C. Ma, C. Chang, T. Lin, and Y. J. Yang, "Highly sensitive tactile sensing array realized using a novel fabrication process with membrane filters," *Journal of Microelectromechanical Systems*, vol. 24, no. 6, pp. 2062–2070, Dec 2015.

[12] M. Xie, K. C. Aw, and W. Gao, "Skin force sensor using piezoresistive pedot:pss with arabitol on flexible pdms," in *2015 IEEE SENSORS*, Nov 2015, pp. 1–4.

[13] M. Shimojo, A. Namiki, M. Ishikawa, R. Makino, and K. Mabuchi, "A tactile sensor sheet using pressure conductive rubber with electrical-wires stitched method," *IEEE Sensors Journal*, vol. 4, no. 5, pp. 589–596, Oct 2004.

[14] J. Cheng, M. Sundholm, B. Zhou, M. Hirsch, and P. Lukowicz, "Smart-surface: Large scale textile pressure sensors arrays for activity recognition," *Pervasive and Mobile Computing*, vol. 30, pp. 97 – 112, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119216000092

[15] W. E. Snyder and J. St. Clair, "Conductive elastomers as sensor for industrial parts handling equipment," *IEEE Transactions on Instrumentation and Measurement*, vol. 27, no. 1, pp. 94–99, March 1978.

[16] L. Xu, G. Chen, J. Wang, R. Shen, and S. Zhao, "A sensing cushion using simple pressure distribution sensors," in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 2012, pp. 451–456.

[17] H. Wang, D. Zhou, and J. Cao, "Development of a skin-like tactile sensor array for curved surface," *IEEE Sensors Journal*, vol. 14, no. 1, pp. 55–61, Jan 2014.

[18] Y. Ding, J. Yang, C. R. Tolle, and Z. Zhu, "Flexible and compressible pedot:pss@melamine conductive sponge prepared via one-step dip coating as piezoresistive pressure sensor for human motion detection," *ACS Applied Materials & Interfaces*, vol. 10, no. 18, pp. 16 077–16 086, 2018, pMID: 29651841. [Online]. Available: https://doi.org/10.1021/acsami.8b00457

[19] H.-B. Yao, J. Ge, C.-F. Wang, X. Wang, W. Hu, Z.-J. Zheng, Y. Ni, and S.-H. Yu, "A flexible and highly pressure-sensitive graphenepolyurethane sponge based on fractured microstructure design," *Advanced Materials*, vol. 25, no. 46, pp. 6692–6698, 2013. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201303041

[20] J. Meyer, B. Arnrich, J. Schumm, and G. Troster, "Design and modeling of a textile pressure sensor for sitting posture classification," *IEEE Sensors Journal*, vol. 10, no. 8, pp. 1391–1398, Aug 2010.

[21] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–, [Online; accessed May 20, 2019]. [Online]. Available: http://www.scipy.org/

[22] C. T. Kelley, "Numerical methods for nonlinear equations," *Acta Numerica*, vol. 27, p. 207287, 2018.

[23] B. W. Lee and H. Shin, "Feasibility study of sitting posture monitoring based on piezoresistive conductive film-based flexible force sensor," *IEEE Sensors Journal*, vol. 16, no. 1, pp. 15–16, Jan 2016.

[24] D. A. Valle-Lopera, A. F. Castaño Franco, J. Gallego-Londono, and A. M. Hernńdez-Valdivieso, "Test and fabrication of piezoresistive sensors for contact pressure measurement," *Revista Facultad de Ingeniería Universidad de Antioquia*, pp. 47 – 52, 03 2017.

[25] S. Salibindla, B. Ripoche, D. T. H. Lai, and S. Maas, "Characterization of a new flexible pressure sensor for body sensor networks," in *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, April 2013, pp. 27–31.

[26] H. Liu, X. Xie, M. Millar, M. Edmonds, F. Gao, Y. Zhu, V. J. Santos, B. Rothrock, and S. Zhu, "A glove-based system for studying hand-object manipulation via joint pose and force sensing," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 6617–6624.

[27] J. Tolvanen, J. Hannu, and H. Jantunen, "Hybrid foam pressure sensor utilizing piezoresistive and capacitive sensing mechanisms," *IEEE Sensors Journal*, vol. 17, no. 15, pp. 4735–4746, Aug 2017.