Touch Position Detection in Electrical Tomography Tactile Sensors Through Quadratic Classifier

Stefania Russo, Roy Assaf, Nicola Carbonaro, Alessandro Tognetti

Abstract—Traditional electrical tomography tactile sensors consider the usage of the system's finite element model. This approach brings disadvantages that jeopardise their applicability aspect and wide use. To address this limitation, the main thrust of this work is to present a method for touch position identification for an electrical tomography flexible tactile sensor. This is done by using a supervised machine learning algorithm for performing classification, namely quadratic discriminant analysis. This approach provides accurate contact location identification, increasing the detection speed and the sensor versatility when compared to traditional electrical tomography approaches. Results obtained show classification accuracy rates of up to 91.6% on unseen test data and an average euclidean error ranging from 1 to 10 mm depending on the contact location over the sensor. The sensor is then applied in real case scenarios to show its efficiency. These outcomes are encouraging since they promote the future practical usage of these type of flexible and low-cost sensors.

Index Terms—Electrical Tomography, Flexible Sensors, Machine Learning, QDA, Classification

1 INTRODUCTION

IN the field of Human–Robot Interaction (HRI), there is an increasing need for unobtrusive sensing devices that do not interfere with the robot's mechanics. [1], [2]. In the last years, many sensor designs for application on robots have been presented in literature as in [3], [4]. These devices are not inherently compliant as they usually consist of discrete and rigid sensing materials embedded in a soft substrate, which creates fragility and stiffness. Therefore, compared to traditional rigid sensors, flexible/stretchable sensors are attracting considerable interest. These can be highly compliant with the host system, while still providing good impact resistance, a reduction in costs, and tolerance to strains [5]. Yet the way these sensors are meant to be embedded within the physical systems is still a challenging task. In fact, existing flexible tactile sensor prototypes often present low accuracy and reliability, which makes it difficult to effectively utilise them in many tasks. Also, the presence of internal structures (i.e. wires, flexible Printed Circuit Boards (PCBs)) within the sensing area still makes them difficult to integrate without interfering seamlessly with the host's mechanics [6]. These disadvantages become more serious when considering safety issues, and incur additional maintenance costs.

An approach that has been recently used to compensate for this drawback, and which is the main topic of this work, is electrical tomography [7]. In a typical tomographic application, conducting surface electrodes are attached around the body part being examined. Current is applied to some electrodes, the resulting potentials being recorded from the

other electrodes. This process is then repeated for numerous different electrode configurations and finally results in a two-dimensional reconstruction of the internal conductivity of the body from boundary voltage data [8].

In tactile sensing applications, this technique allows to place the electrodes only on the boundary of the active sensing area of the sensor, eliminating all internal wiring. As a consequence, electrical tomography-based sensors can be ideally placed over different surfaces and robotic joints, allowing them to easily move.

In our previous studies [9], [10] we have shown that it is possible to apply electrical tomography on soft piezoresistive fabrics that respond to a mechanical solicitation by changing their internal resistance, and realise large area sensor patches, without internal wires.

However, electrical tomography is an inverse problem and mathematically severely ill-posed and non-linear [8]. In typical tomography imaging, the inverse problem is traditionally solved through constructing a Finite Element (FE) model of the system, which subsequently allows for the reconstruction of the internal conductivity of the body under examination. Relying on the FE model for solving the inverse problem has high computational costs and greatly deteriorates the time resolution of the system [11], [12]. This is because the image reconstruction rate heavily depends on the computation time of the inverse solution. Also, as a consequence of using this approach, the reconstruction results are greatly influenced by: small variations in the measured data caused by electrical noise and electrode movements; and modelling errors which include contact impedance, electrode size, boundary shape of the sensor, and electrode misplacement. This is reported in the literature by many authors, see for example [8], [13], [14], [15]. This negative effect becomes particularly visible in the context of electrical tomography-based flexible sensors,

- S. Russo and R. Assaf are with the School of Computing Sciences and Engineering, University of Salford, M5 4WT, UK.
E-mail: s.russo@edu.salford.ac.uk
- N. Carbonaro and A. Tognetti are with the Information Engineering Department and the Research Center "E. Piaggio", University of Pisa, Italy

where modelling errors can be introduced when placing the sensor over curved, irregular or moving surfaces. For example, the boundary electrodes might be moved to positions that were not expected in the initial system model. This creates reconstruction errors in touch input position detection. Therefore, such drawbacks hinder the application of these sensors in real world scenarios. Consequently, although traditional approaches to tomographic imaging are robust and well studied, their disadvantages stand in the way when considering the applicability aspect and wide use of flexible tactile sensors.

Machine Learning (ML) is a branch of Artificial Intelligence (AI) that focuses on learning from data [16], [17]. ML strategies are used in tactile sensing applications [18], [19], [20] due to their ability of extracting meaningful tactile information when the underlying sensing phenomenon is particularly complex.

In the case of tomography systems, ML approaches can be generally divided in three main areas: approaches that use ML techniques to solve the tomography reconstruction inverse problem; ML techniques used to classify the reconstructed tomographic images; and ML approaches that use tomography raw data to extract information regarding the input stimuli. This is shown in Figure 1.

Fig. 1: Machine learning techniques in tomography systems

Reconstruction algorithms based on Artificial Neural Networks (ANN) or Particle Swarm Optimisation (PSO) are common examples of ML techniques used to solve the tomography reconstruction inverse problem [21], [22]. While these approaches present good results, they are still sensitive to noise. In the second approach, tomographic reconstructed images are used as features during the training of the ML algorithms. In the literature, one example of this approach is shown in [23] to classify eight different types of touch or in [24] for hand gesture analysis. However, this approach still relies on solving the tomography inverse problem by using the FE physical model of the sensor, keeping the drawbacks of high sensitivity to electrode misplacement and to noise that have been discussed earlier. Finally the last method, which is the one used in this work, applies ML directly on tomographic raw data. Here there is no need for generating an image that has to be further interpreted by the learning algorithm. Therefore, effective results with minimal computational requirements can be achieved. For example, in [25], the authors present an approach based on ML and tomographic scanning that can be applied to electri-

cally conductive objects. A Convolutional Neural Network (CNN) architecture is used in [26], where different spatio-temporal gaits are classified from tomography raw sensor data from optical fibres.

The aim of the work presented here is to address the limitations found in traditional electrical tomography flexible sensors by proposing an approach for touch position identification based on ML. We present our flexible tactile sensor based on electrical tomography and we propose to identify tactile contact location using a ML algorithm, namely discriminant analysis. This is used to classify voltage data collected by our sensor system. This approach does not rely on the system FE model, thus it provides accurate touch location identification, improves detection speed and promotes sensor versatility when compared to traditional electrical tomography approaches. As a result, the application of these sensors in real world scenarios is also promoted, as later shown in this manuscript. To the best of our knowledge, this is the first time such an approach is used in the context of electrical tomography-based flexible sensors. Also, we contribute to the literature by providing a framework for the easy usage of these sensors.

This paper is organised as follows: we first present our design goals and the sensor fabrication. We then focus on discriminant analysis and propose our novel methodology for data collection and process for robust training of the learning algorithm. This is performed in order to reduce the effects of small dynamic issues that are especially found in piezoresistive materials when applying variable pressure inputs. Then, driven by the aim of providing a framework for a practical and versatile sensor, we present a detailed study on how the system performance are dependent from parameters as: number of training data; number of electrodes; dimension of the sensor; and distance of the touch input from the electrodes. The sensor is finally applied in real case scenarios over different surfaces to show and validate its efficiency. Our conclusions and future remarks are drawn in the final section of the work.

2 METHODS

2.1 Design Goals

Our aim is to develop an electrical tomography sensor system that is practical to implement in real world scenarios. We especially consider the possible utilisation of the tactile sensor in human-interactive applications (e.g. whole-body-touch sensation for humanoid robots [27], safe physical HRI, or flexible touch screens [28]).

Therefore, we determine the necessary specifications of our tactile flexible sensor system as follows.

- The main objective of the sensor is to provide accurate positions for contacts on its surface.
- Essential requirements for the fabric sensor are low cost and the ability to be manufactured with commonly available tools and materials. For good integration on arbitrarily shaped surfaces, the material must be flexible, thin and light weight. Also, it should be possible to customise the shape and spatial resolution of the sensor.
- The training process for contact location identification should be easy and quick.

- The sensor needs to be able to detect touch contacts as small as the human fingertip, between 10 to 15 mm.
- Next, we consider the working range: the sensor needs to work properly under both medium (10-100 kPa, suitable for object manipulation) and low-pressure regimes (under 10 kPa, comparable to gentle touch [29], [30]).

2.2 Skin fabrication

The fabric sensor used in this work is based on the piezoresistive effect. If pressure is applied to such a piezoresistive material, its electrical resistance decreases. The sensor is manufactured as a combination between 3 materials (Figure 2). A honeycomb mesh is placed between a highly conductive bottom fabric material from Less EFM [31] and a high-resistance piezoresistive fabric material from Eeonyx Corp [32]. By placing the honeycomb mesh, we ensure the top and bottom layer are in contact only when a input touch is applied over the sensor. In fact, the bottom low resistive layer constitutes an impedance reducer. This amplifies the local conductivity change when a touch input is applied to the top sensing layer and assures an improved functioning of the sensor. For the electrodes, we used commonly available clip electrodes placed on the top layer and connected to the data acquisition system through a 3-ply stainless steel thread.

Fig. 2: The 3 fabric layers, as embedded in the flexible sensor. A honeycomb mesh is placed between the piezoresistive fabric (on top) and the highly conductive bottom fabric.

The resulting sensor is flexible and may be cut to arbitrary shapes, can be built with low-cost, widely available materials, and low effort. The total thickness of the sensor is around 1.5mm that can be cut and sewn in the same way as a common fabric, thus allowing a wide variety of final shapes to be produced with low effort. We also manufactured and tested different sensor prototypes which are a 13x13 cm square sensor and a 17x17 cm square sensor. Each sensor is tested using 16 and 8 boundary electrodes.

2.3 Classification via Discriminant Analysis

Our aim is to accurately detect the position of an input touch on the sensor. Here, we approach this problem using ML, specifically using supervised learning for performing classification. This is done by first acquiring data from the sensor. This is the training dataset that will be used to build the model. Each electrode reading is treated as a feature, i.e. the response properties on which we would

like to predict results. We then consider the surface of the sensor as a grid, whereby each cell in the grid points to a specific position on the sensor. These cell components define different touch position classes $c = 1, \dots, C$ where C denotes the total number of classes. Moreover, when working with a sensor system, we usually acquire data with some uncertainty. These occur mainly due to measurement error, and unforeseen changes to the system. It is therefore necessary to account for this uncertainty, and we do so by approaching our classification problem from a probabilistic point of view. This is done by using discriminant analysis, whereby a data observation X is treated as an p -dimensional random vector $X = [X_1, X_2, \dots, X_p]$ which is assumed to be drawn from a multivariate Gaussian distribution, and where p denotes the number of features. We proceed by modelling the distribution of X in each class separately. The multivariate Gaussian density for each class is defined as:

$$f_c(x) = \frac{1}{\sqrt{(2\pi)^p |\Sigma_c|}} \exp\left(-\frac{1}{2}(x - \mu_c)^T |\Sigma_c|^{-1} (x - \mu_c)\right) \quad (1)$$

where μ_c denotes the p -dimensional mean vector, and Σ_c denotes the $p \times p$ covariance matrix.

The two most typical uses of discriminant analysis are Linear Discriminant Analysis (LDA), and Quadratic Discriminant Analysis (QDA) [33]. The difference between these two approaches lies in the shape of the decision boundaries that is used to separate the classes. For LDA and QDA these are linear and quadratic decision boundaries respectively. This difference results from the covariance matrices that are used. If the covariance matrix is shared between all the classes, i.e. $\Sigma_c = \Sigma \forall c$, then we are discussing the case of LDA and we can therefore estimate the parameters of Eq 1 using our training data and the following:

$$\hat{\mu}_c = \frac{1}{n_c} \sum_{i=1}^{n_c} x_i \quad (2)$$

$$\hat{\Sigma} = \frac{1}{n - c} \sum_{c=1}^C \sum_{i; y_i=c} (x_i - \hat{\mu}_c)(x_i - \hat{\mu}_c)^T \quad (3)$$

where n_c denotes the number of observations in each class c , and n denotes the total number of observations in the training data. In the case of QDA the difference is that the covariance matrix is instead estimated for each class separately. Afterwards, we use Bayes theorem to obtain the posterior $Pr(Y|X)$, where Y represents a random variable that takes on a specific class c .

An objective function $\delta_c(x)$ can then be derived from [34]. Each new data is finally classified by using $\hat{c} = \operatorname{argmax}_c \delta_c(x)$. The decision boundary between each pair of classes c_i and c_j is then described by $\{x : \delta_i(x) = \delta_j(x)\}$.

2.4 Electrical tomography protocol

In a generic electrical tomography protocol, an electrode pair is chosen for current injection, in the meantime electrode pairs are systematically chosen for performing voltage measurements; this scanning process is repeated until every electrode pair has served for the current injection. In this work, we employ the opposite protocol for current injection and voltage measurements, in Figure 3 and treat the boundary electrodes readings as features for the training of our

learning algorithm. Here, the current injecting electrodes are 180° apart, while adjacent electrode pairs are used for performing voltage readings [35]. This approach increases the current spread inside the material, but also reduces the number of independent measurements due to the reciprocity principle [36]. Whereas this is usually considered as a drawback in electrical tomography [37], in our case it results instead in a dimensionality reduction, since the boundary measurements are treated here as features. As a consequence, when training the ML classifier complexity and training time of the classification model are reduced. This has direct implications in the specific case of QDA, where the covariance matrices are of size $p \times p$.

Using an opposite protocol, the total number of features is $p = L(L - 4)/2$, where L stands for the number of electrodes. Therefore when considering 16 and 8 electrode sensor systems, we have $p = 96$ and $p = 32$ respectively.

Fig. 3: A typical electrical tomography current injection and voltage measurement cycle in the opposite pattern.

2.5 Data collection and Feature Extraction

The experiments presented here are performed using our custom PCB, together with the DAQ and a PC workstation, as described in our previous work [9]. A flat circular target representing the human fingertip is used to apply a touch input over the classes to be tested. Two different targets of size 15 mm and 10 mm were used during the experiments resulting in respectively 49 and 100 classes, as shown in Figure 4. This is performed for the two different sensor dimensions and in the case of 16 and 8 electrodes. The target classes are precisely defined on the sensor by placing a paper grid on the fabric.

Fig. 4: On the left, touch inputs for a 1.5 cm target (49 classes) and on the right, a 1 cm target (100 classes) in the case of 16 electrodes.

2.5.1 Training data

In this section, we show how we collect the observations for each class and extract the necessary features for the learning algorithm to be trained.

During the experiments, for each touch input presented to the sensor, we acquire $S_{num}=20$ number of scans. A scan is a full cycle of current injections and voltage measurements throughout all the electrodes. $S=3200$ samples are collected for each scan at 10kHz sampling rate. We have found that this number of samples guarantees a good compromise between data acquisition time and quantity of data. First, a small weight is placed on top of the target to guarantee that a minimum pressure of approximately 10 kPa is achieved. Then, the applied pressure over each class is varied manually. At this step we are not searching for a high precision in the applied pressure because we want to have a high variety of observations for the training dataset. This allows the resulting trained ML algorithm to identify touch input locations independently of their pressure values. The purpose behind this choice is that we are trying to cover a certain range of pressure range as described in Section 2.1. Then, after S_{num} scans are completed, the target is applied on the next class and the process is repeated until all the input classes are tested. The full process is finally repeated for $T = 6$ times. The entire data acquisition takes about 1 hour.

In order to form the training matrix, once the raw data are collected, we then divide the samples in S_{sub} subsections of 160 samples each and perform the median, as described in our previous work [9]. This step provides further robustness to the classification algorithm as it is able to train on a high variety of data, giving stability to noise. Also, this way of acquiring and engineering training data allows the learner algorithm to take into account small dynamic issues as elastic return, or hysteresis effect that are present between the application of variable pressures. Finally, the number of observations n_c in the training matrix for each class becomes:

$$n_c = TS_{num}S_{sub}. \quad (4)$$

In our case, $n_c = 2400$. The process described above is reported in Algorithms 1 and 2. This training technique can be easily performed by any user, without requiring sophisticated hardware.

2.5.2 Test data

After the ML algorithm is trained using the process described above, it needs to be tested with unseen data. 20 test data for each class were acquired 1 day after the training process. Each test data is acquired following [9] and is a touch input over the sensor with an unknown pressure. This is performed to test and show that the final ML algorithm is able to identify touch input locations independently of their pressure values.

2.6 Training process

During the training process of the ML algorithm, a segment of validation data set is divided from the training data and used to validate the model. This estimates how well the model has been trained and its properties.

Algorithm 1: Training data algorithm, first part

S number of collected samples per scan;
 S_{num} number of scans per class;
 C number of position classes;
 T number of acquisitions;
 L number of electrodes;
 S_{sub} constant used for dividing samples into sub sections;

input : $1 \times (C * T)$ vector container $data_{con}$ of all readings, where each element consists of a $(S_{num} * S) \times L$ measurement matrix
output: $1 \times C$ vector container $median_{cont}$, where each element represents a class, and consists of a $(S_{sub} * S_{num} * T * L) \times L$ matrix of sample medians

Stacking tests per class;
for $j \leftarrow 1$ **to** C **do**
 $cnt \leftarrow 1$;
 for $j \leftarrow i$ **to** $C * T$ **by** C **do**
 $data_{pc}\{cnt, j\} \leftarrow C\{i\}$;
 $cnt \leftarrow cnt + 1$;
 end
end
Computing a container of median values;
 $div_{ratio} \leftarrow S/S_{sub}$;
for $j \leftarrow 1$ **to** C **do**
 $tmp3 \leftarrow \text{empty}$;
 for $i \leftarrow 1$ **to** T **do**
 $tmp1 \leftarrow \text{empty}$;
 $tmp2 \leftarrow \text{empty}$;
 for $k \leftarrow 1$ **to** $S_{num} * S$ **by** div_{ratio} **do**
 $tmp \leftarrow \text{compute column wise median of } data_{pc}\{i, j\}(k \text{ to } k + div_{ratio} - 1)$;
 $tmp2 \leftarrow \text{stack } tmp \text{ horizontally}$;
 end
 $tmp3 \leftarrow \text{stack } tmp2 \text{ horizontally}$;
 end
 $median_{cont}\{j\} \leftarrow tmp3$;
end

Algorithm 2: Training data algorithm, second part

S number of collected samples per scan;
 C number of position classes;
 T number of acquisitions;
 S_{num} number scans per class;
 k_{select} logical vector for selecting useful boundary measurements;
 S_{sub} constant used for dividing samples into sub sections;

input : $1 \times L$ vector container $median_{cont}$, where each element represents a class, and consists of a $(S_{sub} * S_{num} * T * L) \times L$ matrix of sample medians
output: $(C * T * S_{num} * S_{sub}) \times (k_{select}/2)$ matrix used for training the classifier ml_{mat}

Computing a container of features;
for $j \leftarrow 1$ **to** C **do**
 $tmp1 \leftarrow \text{empty}$;
 $tmp2 \leftarrow \text{empty}$;
 for $i \leftarrow 1$ **to** S_{sub} **do**
 for $k \leftarrow 1$ **to** $S * S_{num}$ **by** S_{sub} **do**
 $tmp1 \leftarrow \text{stack } median_{cont}\{j\}(k, all) \text{ vertically}$;
 if $length(tmp1)$ is equal to L^2 **then**
 $tmp2 \leftarrow \text{stack } tmp1 \text{ horizontally}$;
 $tmp1 \leftarrow \text{empty}$;
 end
 end
 end
 $feature_{cont}\{j\} \leftarrow tmp2(all, k_{select}/2)$;
end
Computing the machine learning matrix;
for $j \leftarrow 1$ **to** C **do**
 $tmp \leftarrow \text{a } (S_{sub} * S_{num} * T) \times 1 \text{ column vector of } 1 * j$;
 $tmp \leftarrow \text{stack } feature_{cont}\{j\} \text{ vertically}$;
 $ml_{mat} \leftarrow \text{stack } tmp \text{ horizontally}$;
 $tmp \leftarrow \text{empty}$;
end

When having adequate training data, the validation segment is expected to be a good statistical representation of the entire data set. If not, the results will greatly depend on how the data was divided.

To avoid this, in this work we have used cross validation. Each time, during the training phase, the data is partitioned into k-folds, one of this is held-out for validation, the remaining are used for training. This process is then repeated k times until all the folds are used. After, the k-fold average loss is from all the folds is computed, and is used to efficiently compare the different learning algorithms. In this work, a 10-fold cross-validation is used, as it is the most common approach in data mining and machine learning [38].

It is worth mentioning that a high number of features in the data affects the training time and memory consumption, creating computationally intensive learning algorithms. Using an opposite protocol for current injection and voltage

measurement reduces the number of features. Therefore our approach brings significant benefits during the training process and also results in a simple ML algorithm [39].

To sum up, the training and test data used in this work consist of: $p = 96$ features in the case of a sensor presenting $L = 16$ electrodes, and $p = 32$ features in the case of a sensor with $L = 8$ electrodes; $C = 49$ and $C = 100$ classes in the case of a target size of 1.5 cm and 1 cm respectively; $n_c = 2400$ observations for each class for training; 20 observations for each class for testing.

Our framework for training and classification for touch input identification is shown in Figure 5.

2.7 Average Euclidean Error

In this work we are dealing with a physical sensor system where the separation between each class is a 2-D distance.

Fig. 5: Proposed approach for touch position identification using ML

Therefore, only considering the training and test data classification errors might not be enough to judge the quality of the ML algorithm. This is particularly meaningful in applications where it is acceptable to misclassify two neighbour classes. Euclidean error is a good way to weight the cost associated to each miss-classified class by considering its 2-D distance from the true response class.

We calculate the Euclidean Error EE_{tr} for each true class tr as the euclidean distance between the true class and the other classes, times the number of misclassifications. Then, given the number of classes C , we compute the Average Euclidean Error AEE as:

$$AEE = \frac{\sum EE_{tr}}{C} \quad (5)$$

3 RESULTS

3.1 Classification Methods

Before conducting our work, we have compared the performance of different learning algorithms because of their wide usage in the ML literature: Classification Trees, Support Vector Machines (SVM), K-Nearest Neighbours (KNN) and Ensemble methods. This is done in terms of training accuracy Tr_a and training time Tr_t . This is performed in the case of different numbers of training observations for each class: $n_c = 24$, $n_c = 240$ and $n_c = 2400$. The training observations data used to make this comparison are related to the tests conducted on $C = 49$ classes for the sensor with

ML Algorithm	$n_c = 24$		$n_c = 240$		$n_c = 2400$	
	Tr_a	Tr_t	Tr_a	Tr_t	Tr_a	Tr_t
Classification Trees	78.1%	3	78.8%	17	72%	74
LDA	92.1%	92	94%	9	93.8%	88
QDA	66.8%	2	99%	10	100%	81
SVM Linear	83.8%	52	99.5%	257	98.2%	2761
SVM Quadratic	92.8%	156	99%	302	99.1%	3247
KNN Classification	94.1%	181	99%	304	100%	7422
Ensemble Boosted Trees	68.8%	253	71.6%	555	73.3%	8500
Ensemble Bagged Trees	92.7%	244	99.9%	414	100%	8172
Ensemble Subspace Discriminant	91.2%	251	91.1%	412	91.2%	8172

TABLE 1: Training accuracy and training time (in seconds) for different ML classification algorithms

Electrode Number	Sensor Size	Target Size	Te_a
16	13x13	1.5cm	91.6%
		1cm	77.8%
	17x17	1.5cm	88.7%
		1cm	80.6%
8	13x13	1.5cm	68.98%
		1cm	29.9%
	17x17	1.5cm	56.8%
		1cm	37.4%

TABLE 2: Test accuracy Te_a for different number of electrodes and sensor sizes

size 13x13 cm. The results of the comparison are shown in Table 1. The results show that LDA and QDA present the best performance in terms of training accuracy and time when presented with enough training data. In the other tests, their accuracy can be still considered satisfying. Also, they show short training time. The other learning algorithms are either too slow, with up to 2.5 h training time in the case of $n_c = 2400$, or less efficient, as for example classification trees. In the next sections however, we show that the AEE on the test data is negatively affected by the number of training data already when $n_c = 240$. Therefore, using $n_c = 24$ is in reality of no practicality when presenting the ML algorithm with unseen data. Based on these results, in this work we have applied QDA.

3.2 Test Accuracy

First, in order to characterise the response of the QDA learner algorithm in different scenarios, in Table 2 we report its test accuracy Te_a when changing the number of electrodes and sensor sizes. Is it visible that Te_a is notably lower for the sensor tested with 8 electrodes, and decreases down to 37.4% in the case of sensor and target size of 17x17 cm and 1 cm respectively.

3.3 Number of Training Data

Previously, we have compared the results of training accuracy for different learner algorithms when changing the number of training observations for each class. However, it is interesting to know how the *AEE* changes depending on the number of training data n_c . The *AEE* is calculated following Eq. 5. For the sake of clarity, we only present here the results the sensor with size 13x13 cm with a 1.5 cm target in the case of $L = 16$ and $L = 8$ electrodes. In fact, we believe it is unnecessary to run a full test considering all our variables (sensor size, target size, number of electrodes, training data). The results are shown in Figure 6 and use 3 sets of observation from the test data: $n_c = 2400$, $n_c = 1200$ and $n_c = 240$. It is visible from the figure that when employing less than $n_c = 2400$, the error increases to almost twice its value at $n_c = 2400$.

Fig. 6: Average euclidean error function of different training data

3.4 Distance from the Electrodes

AEE can be used as a metric to show how the distance between each class and the boundary electrodes affects the quality of the touch input identification. For better visualisation, in Figure 7a the various classes are grouped with different colours based on their distance from the electrodes. The *AEE* for each distance group is computed and shown in Figure 7b. We note that the *AEE* tends to increase and reaches up to 10 mm the more the classes are far from the electrodes. This behaviour demonstrates an error that is not stable all over the whole sensing area and depends on the position of the target on the sensor. The *AEE* values range in fact from 1 to 10 mm. This is a drawback in electrical tomography sensors, where the information content tends to decrease when the target is far from the electrodes, as a result of a reduction in the SNR. However, it is worth noticing that in Figure 7b, the error slightly decreases in the most central region of the sensor. This is due to a higher current density flowing in that region. This result confirms our work presented in [10] and opens new future directions on how to tune the response of the classification based on the current density patterns.

3.5 Sensor Size

The *AEE* is calculated in the case of our two different tested sensor sizes. The results are shown in Figure 8. From the

Fig. 7: On top, representation of the various distances from the electrodes for the two different tested sensor sizes. On the bottom, Average Euclidean Error function of the distance from the electrodes.

results, it is important to note that the *AEE* does not change much in relation to the sensor size. These results show that, for studying the sensor's performance, it is in fact more meaningful to consider the distance between the classes and the electrodes (Figure 7), and not the whole sensing area. In fact, by analysing the sensitivity of the error with respect to the distance from the electrodes, we obtain a more meaningful understanding of the sensor's behaviour. This is useful when dealing with different and non symmetrical sensor shapes.

Fig. 8: Average euclidean error and standard deviation when increasing the sensor size.

3.6 Number of Electrodes

The *AEE* is calculated in respect of the number of electrodes considering our two tested target and sensor sizes, and it is shown in Figure 9. Here, the error decreases considerably when switching from $L = 8$ to $L = 16$ electrodes, for all the considered experiments. This result was already expected. This is because when using only

8 electrodes for performing current injection and voltage measurement, there is a great decrease of information for the learning algorithm, given by the reduced features from $p = 96$ to $p = 32$.

4 APPLICATIONS ON REAL SYSTEMS

In this section, we apply our proposed approach in two real world scenarios. The sensor setup and the ML classification algorithm have been implemented in two cases: when the sensor is wrapped around a curved surface; and the when sensor is placed around a robotic arm.

Fig. 9: Average euclidean error when changing the number of electrodes.

4.1 Application over a Curved Surface

The sensor is implemented over a curved surface (Figure 10) that has bending radius of 3 cm. Training data from $C = 14$ touch input classes are collected directly on the curved surface following the procedure explained earlier. Then, QDA algorithm is trained resulting in a training classification accuracy $Tr_a = 100\%$. After that, sensor data is acquired at 30 Hz sensing rate and presented to the trained classifier. The results are shown in real-time on a PC screen in the form of a matrix and can be seen in Figure 11. Each time a touch input class is recognised, the section of the matrix corresponding to that class is illuminated with a different colour. The results confirm that the sensor is able to perform efficiently when bent over a curved surface.

4.2 Application over a Robotic Arm

The sensor is implemented over a robotic arm. The system is proposed in Figure 12 and comprise of: the hardware setup described in [9] with our custom PCB and a DAQ card for current injection and voltage readings, and a workstation PC for data analysis; a Kuka KR10 robotic arm which can be controlled via Matlab commands; and a 22x10 cm fabric flexible sensor with $L = 16$ electrodes. The sensor is intentionally placed with some discontinuities, to test the effective robustness of the approach.

We conduct two experiments explained below.

- 1) Experiment 1: training data from $C = 5$ classes are collected with the procedure explained earlier. Then, QDA algorithm is trained resulting in a training classification accuracy $Tr_a = 99.7\%$. The 5 classes that

Fig. 10: Electrical tomography flexible fabric sensor implemented over a curved surface with bending radius of 3 cm, and presenting $C = 14$ touch input classes

Fig. 11: Results from test data. The trained algorithm is able to recognise correctly the different input classes.

have been used for training are: 1 no-touch class and 4 touch classes over the sensor. After training, sensor data is acquired at 30 Hz sensing rate and presented to the trained classifier. The results are shown in real-time on a PC screen in the form of a matrix. In Figure 13, a full light blue matrix indicates no touch recognised. Then each time a touch input class is recognised, the section of the matrix corresponding to that class is illuminated with a different colour. The results show that the sensor is able to perform efficiently when bent over a curved surface and is able to recognise no-touch inputs.

Fig. 12: Electrical tomography flexible fabric sensor implemented over a robotic arm.

Fig. 13: Identification of no-touch and 4 touch inputs over the Kuka robotic arm

Fig. 14: The kuka robot moving to four different locations on its workbench, being controlled by 4 touch inputs

- 2) Experiment 2: The Kuka robot is controlled using $C = 4$ touch inputs, which are used to send the robot to four different locations on its workbench, this is shown in Figure 14. A video is also available as Supplementary Materials. The classification output is sent to the Kuka robot, where the inverse kinematics are computed to allow it to achieve the desired behaviour. The results show that the sensor can be easily placed over a robotic arm without losing its functionality, and without impeding the robot's dexterity.

5 DISCUSSION AND CONCLUSION

In this work, we have proposed a state of the art method for accurately identifying touch positions using electrical tomography sensors. This is achieved using supervised machine learning for performing classification, namely quadratic discriminant analysis. This methodology increases touch input detection speed. In fact, traditional electrical tomography techniques require a construction of a FE model of the sensing element for the reconstruction of the internal conductivity of the sensor. This greatly deteriorates the time resolution of the system, as already reported in [7], [8], [40]. On the other hand, our approach potentially reaches a detection speed of 78Hz which is our system's sensing rate [9]. This is due to the fact that once the classification algorithm is trained, the computation time for classifying new data collected by the sensor is almost instantaneous.

In addition, our approach demonstrated accurate contact location identification. When applied over curved surfaces, the flexible sensor system was also able to robustly deal with problems such as inaccurate sensor placement, electrode

discontinuities, movements of the host. This overcomes some of the major limitations of electrical tomography tactile sensors.

As a consequence of the results shown in this work, and driven by the aim of proving a framework for a practical and easy to customise flexible sensor, we can conclude that:

- ML techniques can provide accurate contact location identification, and increase the detection speed and sensor versatility when compared to traditional electrical tomography sensor.
- If developing arbitrary sensor shapes, it is suggested to particularly focus on the maximum distance from the furthest touch input class to the electrodes, rather than just on the sensor size. In our experimental results, when a class exceeds a distance from the electrodes of 6-7 cm, the touch identification error increases greatly.
- The above behaviour confirms the idea that in electrical tomography sensors the error is not stable throughout the whole sensing area. Practitioners designing tactile sensors based on electrical tomography need to be aware of this.
- In general, a number of electrodes $L = 16$ is desirable for reaching a good compromise between model complexity, data collection time and good results.
- The average error in touch input identification can be as small as 1 mm when choosing the correct design parameters. However, when the sensor and system designs are not correct, these errors can reach up to 31 mm.

Some challenges and issues remain as an open research topic. This opens up perspectives that can be addressed in future works. In particular, we believe that this work can be further improved by integrating the capability to absolute positioning, and the detection of multiple inputs. This can be achieved using regression techniques.

ACKNOWLEDGMENTS

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement number 608022

REFERENCES

- [1] R. S. Dahiya and M. Valle, *Robotic tactile sensing: technologies and system*. Springer Science & Business Media, 2012.
- [2] R. S. Dahiya, P. Mittendorf, M. Valle, G. Cheng, and V. J. Lumelsky, "Directions toward effective utilization of tactile skin: A review," *IEEE Sensors Journal*, vol. 13, no. 11, pp. 4121–4138, 2013.
- [3] A. Cirillo, F. Ficuciello, C. Natale, S. Pirozzi, and L. Villani, "A conformable force/tactile skin for physical human-robot interaction," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 41–48, 2016.
- [4] T. P. Tomo, S. Somlor, A. Schmitz, L. Jamone, W. Huang, H. Kristanto, and S. Sugano, "Design and characterization of a three-axis hall effect-based soft skin sensor," *Sensors*, vol. 16, no. 4, p. 491, 2016.
- [5] D. Vogt, Y. Menguc, Y.-L. Park, M. Wehner, R. Kramer, C. Majidi, L. Jentoft, Y. Tenzer, R. Howe, and R. J. Wood, "Progress in soft, flexible, and stretchable sensing systems," in *Proceedings of the International Workshop on Research Frontiers in Electronics Skin Technology at ICRA*, vol. 13, 2013.
- [6] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing—from humans to humanoids," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 1–20, 2010.

- [7] D. Silvera-Tawil, D. Rye, M. Soleimani, and M. Velonaki, "Electrical impedance tomography for artificial sensitive robotic skin: a review," *IEEE Sensors Journal*, vol. 15, no. 4, pp. 2001–2016, 2015.
- [8] D. S. Holder, *Electrical impedance tomography: methods, history and applications*. CRC Press, 2004.
- [9] S. Russo, S. Nefti-Meziani, N. Carbonaro, and A. Tognetti, "Development of a high-speed current injection and voltage measurement system for electrical impedance tomography-based stretchable sensors," *Technologies*, vol. 5, no. 3, p. 48, 2017.
- [10] —, "A quantitative evaluation of drive pattern selection for optimizing eit-based stretchable sensors," *Sensors*, vol. 17, no. 9, p. 1999, 2017.
- [11] D. S. Tawil, "Artificial skin and the interpretation of touch for human-robot interaction," Ph.D. dissertation, School Aersp., Mech. Mechatron., Univ. Sydney, New South Wales, Australia, 2012.
- [12] A. Boyle, A. Borsic, and A. Adler, "Addressing the computational cost of large eit solutions," *Physiological measurement*, vol. 33, no. 5, p. 787, 2012.
- [13] Y. Asfaw and A. Adler, "Automatic detection of detached and erroneous electrodes in electrical impedance tomography," *Physiological Measurement*, vol. 26, no. 2, p. S175, 2005.
- [14] V. Kolehmainen, M. Vauhkonen, P. Karjalainen, and J. Kaipio, "Assessment of errors in static electrical impedance tomography with adjacent and trigonometric current patterns," *Physiological Measurement*, vol. 18, no. 4, p. 289, 1997.
- [15] E. Gersing, B. Hofmann, and M. Osypka, "Influence of changing peripheral geometry on electrical impedance tomography measurements," *Medical and Biological Engineering and Computing*, vol. 34, no. 5, pp. 359–361, 1996.
- [16] K. Murphy, "Machine learning: a probabilistic approach," *Massachusetts Institute of Technology*, pp. 1–21, 2012.
- [17] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.
- [18] P. Gastaldo, L. Pinna, L. Seminara, M. Valle, and R. Zunino, "A tensor-based approach to touch modality classification by using machine learning," *Robotics and Autonomous Systems*, vol. 63, pp. 268–278, 2015.
- [19] S. S. Baishya and B. Bäuml, "Robust material classification with a tactile skin using deep learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016. IEEE, 2016, pp. 8–15.
- [20] A. Schneider, J. Sturm, C. Stachniss, M. Reiser, H. Burkhardt, and W. Burgard, "Object identification with tactile sensors using bag-of-features," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*. IEEE, 2009, pp. 243–248.
- [21] A. Adler and R. Guardo, "A neural network image reconstruction technique for electrical impedance tomography," *IEEE Transactions on Medical Imaging*, vol. 13, no. 4, pp. 594–600, 1994.
- [22] S. Martin and C. T. Choi, "Nonlinear electrical impedance tomography reconstruction using artificial neural networks and particle swarm optimization," *IEEE Transactions on Magnetics*, vol. 52, no. 3, pp. 1–4, 2016.
- [23] D. Silvera Tawil, D. Rye, and M. Velonaki, "Interpretation of the modality of touch on an artificial arm covered with an eit-based sensitive skin," *The International Journal of Robotics Research*, vol. 31, no. 13, pp. 1627–1641, 2012.
- [24] Y. Zhang and C. Harrison, "Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition," in *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*. ACM, 2015, pp. 167–173.
- [25] Y. Zhang, G. Laput, and C. Harrison, "Electrick: Low-cost touch sensing using electric field tomography," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 1–14.
- [26] O. Costilla-Reyes, P. Scully, and K. B. Ozanyan, "Deep neural networks for learning spatio-temporal features from tomography sensors," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 1, pp. 645–653, 2018.
- [27] P. Mitterdorfer and G. Cheng, "Humanoid multimodal tactile-sensing modules," *IEEE Transactions on robotics*, vol. 27, no. 3, pp. 401–410, 2011.
- [28] J. Wang, M. Liang, Y. Fang, T. Qiu, J. Zhang, and L. Zhi, "Rod-coating: towards large-area fabrication of uniform reduced graphene oxide films for flexible touch screens," *Advanced Materials*, vol. 24, no. 21, pp. 2874–2878, 2012.
- [29] S. C. Mannsfeld, B. C. Tee, R. M. Stoltenberg, C. V. H. Chen, S. Barman, B. V. Muir, A. N. Sokolov, C. Reese, and Z. Bao, "Highly sensitive flexible pressure sensors with microstructured rubber dielectric layers," *Nature materials*, vol. 9, no. 10, p. 859, 2010.
- [30] L. Pan, A. Chortos, G. Yu, Y. Wang, S. Isaacson, R. Allen, Y. Shi, R. Dauskardt, and Z. Bao, "An ultra-sensitive resistive pressure sensor based on hollow-sphere microstructure induced elasticity in conducting polymer film," *Nature communications*, vol. 5, p. 3002, 2014.
- [31] "Less EMF Fabrics," <https://www.lessemf.com/fabric.html>, accessed: 2018-03-20.
- [32] J. Avloni, "Electroconductive woven and non-woven fabric," Dec. 23 2008, uS Patent 7,468,332.
- [33] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1.
- [34] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.
- [35] M. Wang, Q. Wang, and B. Karki, "Arts of electrical impedance tomographic sensing," *Phil. Trans. R. Soc. A*, vol. 374, no. 2070, p. 20150329, 2016.
- [36] D. B. Geselowitz, "An application of electrocardiographic lead theory to impedance plethysmography," *IEEE Transactions on Biomedical Engineering*, vol. 18, no. 1, pp. 38–41, 1971.
- [37] X. Shi, X. Dong, W. Shuai, F. You, F. Fu, and R. Liu, "Pseudopolar drive patterns for brain electrical impedance tomography," *Physiological Measurement*, vol. 27, no. 11, p. 1071, 2006.
- [38] P. Refaellizadeh, L. Tang, and H. Liu, "Cross-validation," in *Encyclopedia of Database Systems*. Springer, 2009, pp. 532–538.
- [39] A. L. Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 245–271, 1997.
- [40] A. Nagakubo, H. Alirezai, and Y. Kuniyoshi, "A deformable and deformation sensitive tactile distribution sensor," in *IEEE International Conference on Robotics and Biomimetics, ROBIO 2007*. IEEE, 2007, pp. 1301–1308.