Tactile Data Classification of Contact Materials Using Computational Intelligence

Sergio Decherchi *Student Member IEEE*, Paolo Gastaldo, Ravinder S. Dahiya, *Member, IEEE*,

Maurizio Valle, *Member, IEEE,* and Rodolfo Zunino, *Senior Member, IEEE*

***Abstract*— The two major components of a robotic tactile sensing system are the tactile sensing hardware at the lower level, and the computational/software tools at the higher level. Focusing on the later, this research assesses the suitability of Computational Intelligence tools for tactile data processing. In this context, this paper addresses the classification of sensed object material from the recorded raw tactile data. For this purpose, three computational intelligence paradigms, namely, Support Vector Machine (SVM), Regularized Least Square (RLS) and Regularized Extreme Learning Machine (RELM) have been employed and their performance compared for the said task. The comparative analysis shows that SVM provides the best trade-off between classification accuracy and computational complexity of the classification algorithm. Experimental results indicate that the Computational Intelligence tools are effective in dealing with the challenging problem of material classification.**

***Index Terms*— Computational intelligence, Machine learning, Tactile sensing, Material classification, tactile data processing**

I. Introduction

ADVANCED robotic systems require multi-modal sensory information to interact safely, to take decisions, and to successfully carry out actions – all in an autonomous way. Sensor based robotic interaction has generally been investigated using vision and auditory sensors. However, the information obtained with visual and auditory sensors can sometimes be misleading due to the lack of contact information. The rich interaction behaviors exhibited by real-world objects also depend on how heavy and stiff the contacted objects are; how their surface feel when touched; how they deform on contact and how they move when pushed etc. Therefore using the tactile data along with that coming from existing sensory apparatus will greatly enhance the real world model generation capability of robots [1].

A robotic tactile sensing system consists of two major components. First, the tactile sensing hardware - comprising of tactile sensors and associated interface electronics. Second, the computational tools that help to obtain real-world models or to achieve some prescribed target (e.g. material classification, texture classification etc.) from raw tactile data. In general, the tactile hardware has attracted a greater attention and accordingly large number of tactile sensors have been developed [1, 2]. On other hand, little has been done towards developing and using the computational models or algorithms to interpret the information hidden in the tactile data. This paper analyzes and compares the performance of different Computational Intelligence (CI) paradigms for handling raw tactile data. The rationale behind the proposed approach is that CI algorithms represent a powerful tool to successfully tackle complex, often non-linear [3] problems such as those involved in tactile sensing.

The literature gives a number of works addressing the use of CI algorithms for vision sensing [4]. CI techniques have been also used to analyze touch sensing data for surface roughness classification [5], contact angle recognition [6], inverse touch sensing problems, object primitive detection, texture pattern recognition [7], and material classification [8]. In this paper, CI algorithms are used to address material classification. For this purpose, three well known classification paradigms, Support Vector Machine (SVM) [3], Regularized Least Square (RLS) [9] and an extension of the basic Extreme Learning Machine (ELM) [10], are implemented and their performances compared. The tactile data feeding the classifiers was obtained by a piezoelectric tactile sensor hosted by a chip developed for the fingertips of a humanoid robot [11]. The experiments presented in this paper involve four different materials (i.e. brass, wood, plastic, and lead) that are classified using CI techniques. Experimental results demonstrate that CI algorithms can effectively accomplish the material-classification task.

The paper is organized as follows. Section II presents the tactile sensing hardware employed for recording the data that is suitable for the classifiers. Section III presents the three CI models used to address material classification. Following this, the experimental results are presented in Section IV. Finally, the results are summarized as conclusions in Section V.

II. Measuring The Tactile Data

The tactile data to be used as input to the CI-based classifiers was recorded using a single sensor of high resolution MEA (Microelectrode Array) based tactile sensing chips [11]. This

Manuscript received …

S. Decherchi, P. Gastaldo, M. Valle and R. Zunino are with the Dept. of Biophysical and Electronic Engineering, University of Genoa, Via Opera Pia 11a, 16145, Genova, Italy (e-mail: {sergio.decherchi, paolo.gastaldo, maurizio.valle , rodolfo.zunino}@unige.it).

R. S. Dahiya is with CMM (Bio-MEMS) division of Fondazione Bruno Kessler, Via Sommarive 18, 38123 Trento, Italy (e-mail: dahiya@fbk.eu).

section presents the measurement of the tactile data.

A. The Tactile Sensing Chip

A high resolution MEA based tactile sensing chip (details reported elsewhere [11]) has been used in this work to collect the tactile data. The tactile sensing chip consists of a 2-D array of 32 microelectrodes that are epoxy adhered with piezoelectric polymer (Polyvinylidene Fluoride - Trifluoroethylene). In sensing/generating mode, a force/stress is applied on the piezoelectric polymer. As a result, a proportional charge/voltage is generated. In the present case, the applied force means the contact force between the sensor and the object. When connected to the gate terminal of the FET devices (external to chip), each microelectrode can be considered as an extended gate. In doing so, the charge/voltage generated in the piezoelectric polymer will be reflected in the induced channel of FET devices [2],which can be converted in voltage and processed further with the help of external signal conditioning circuitry. A simple and alternate way of doing the same is to use a charge/voltage amplifier across each microelectrode. This work therefore utilizes a charge amplifier to interface the output of a single tactile sensor of the array with signal acquisition electronics. The scheme of charge amplifier, along with the approximate equivalent model of a MEA with polymer, is explained elsewhere [11]. The applied force is generated by a shaker/vibration generator and the value of the applied force is recorded using a standard piezoelectric force sensor.

B. Tactile Data Measurement for material classification

An interesting feature of a piezoelectric transducer is that its output is influenced by the type of materials present on its front and backside [12]. The origin of such variation in the response lies in the fact that various materials have distinct mechanical impedances. When an object *A*, having mechanical impedance Z_A, comes in contact with the piezoelectric transducer *C*, having mechanical impedance Z_C, the velocity of particles, *u*, in two mediums is different. This is because of different densities and the boundary conditions at the interfaces, which require that the stresses and displacements on both sides of the interface shall be equal**.** The same holds at the other interface (substrate B in Fig. 1) of the piezoelectric transducers where another material with mechanical impedance Z_B can be present. The particle velocity is related with particle displacement and hence with the strain [13, 14]. As strained piezoelectric transducer generates charge, its output must be proportional to difference in the particle velocity and hence to the difference in forces at two interfaces. Thus, if mechanical impedances of materials at two interfaces of piezoelectric transducer are known, the response of piezoelectric polymer can be obtained with the knowledge of forces at two interfaces [14]. This in turn means that if response of piezoelectric transducer is known and if the contact forces at two interfaces are known, then one obtain the mechanical impedance of materials and hence the type of material. Further, if the substrate material is always same, as in the MEA used in this work, the contact force at first

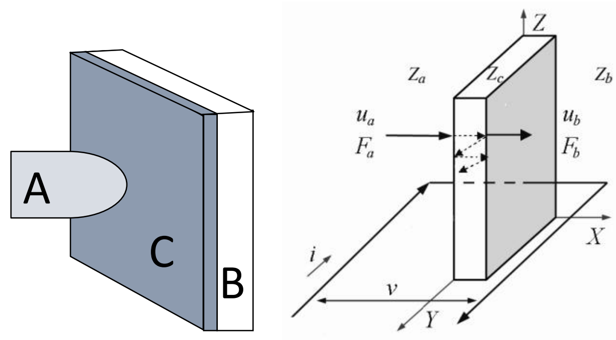

Figure 1: The output of piezoelectric transducer depends on the type of materials present on its front and back side.

interface only is sufficient. For this reason, force amplitude and sensor output amplitude have been chosen as the two features to be processed by the material recognition system.

In addition to the physical contact conditions, various material constants such as piezoelectric constant, elastic constant, and dielectric constant also determine the response of a piezoelectric transducer. In general, these constants are frequency dependent [14]. During contact, the sensor on the robotic platform may come across dynamic contact forces that comprise of a range of frequencies. As a result, the parameters of piezoelectric transducer and hence the output may vary. In addition to this, soft piezoelectric materials such as P(VDF-TrFE) polymers also exhibit, the viscoelastic, dielectric and piezoelectric losses. These losses are represented by using complex values of elastic, piezoelectric and dielectric constant [14]. In view of above factors affecting the sensor output, the frequency and phase of the signal too have been included in the feature space feeding the classification system.

The following procedure was eventually applied to collect data for simulations involving the CI-based classification tools. Probes made of different materials were used for applying the dynamic normal forces (same amplitude used for all probes) on the touch sensors described earlier. In particular, probes with 1 mm diameter, and made from brass (density ~ 8500 Kg/m^3), polycarbonate plastic (density ~ 1200 Kg/m^3), wood (density ~ 500 Kg/m^3), and pencil lead (density ~ 2100 Kg/m^3) were used. For each probe, tactile data were collected by applying the following procedure: the dynamic forces of 0.2 N and 0.4 N were applied on single sensor of the tactile sensing chip at 5, 15, 30, 45, and 70 Hz frequencies and the output was sampled and acquired. Phase shift is worked out by comparing the input and output force sinusoids. The tactile measurement pattern included four measures: input frequency, input force, sensor output phase and amplitude. No pre-processing or feature extraction was applied on the data.

III. MACHINE LEARNING FOR CLASSIFICATION

Machine learning techniques provide effective tools to design predictive systems that make reliable decisions on unseen input samples [3]. This paper focuses on three alternative techniques: Regularized Least Squares (RLS) [9], Support Vector Machines (SVM) [3] (both instances of kernel machines) and Regularized Extreme Learning Machine (an instance of neural networks). The latter paradigm represents a numerically stabilized version of the standard Extreme Learning Machine [10]. RLS, SVM and RELM usually tackle binary classification problems (i.e., two class problems). Actually, other machine learning methods exist that inherently solve multiclass problems (e.g., Bayesian). However, the literature provides effective strategies to address a multiclass classification scheme by exploiting binary classifiers; the present work adopts the "1 vs ALL" strategy [15].

Three main aspects make RLS, SVM and RELM suitable for the present applicative domain. First, these models represent powerful classifiers that already proved very effective in dealing with complex problem domains [3]. Second, they are well suited for hardware implementations. Third, the three models provide different representation paradigms; hence, a comparison of the performances in material recognition between those tools can lead to interesting outcomes that may support future activities.

A. Classification Algorithms for Tactile Data

The training of learning systems requires a dataset, $\boldsymbol{X}$, holding n patterns (samples): each pattern includes a data vector, $\mathbf{x} \in R^m$, and the category label, $y \in \{-1, +1\}$. When developing data-driven classifiers, the learning phase requires both $\mathbf{x}$ and y to build up a decision rule. After training, the system can process data that do not belong to the training set; the system classifies each input sample with a predicted category, $\hat{y}$.

The function that predicts the class of a sample is the decision function, $\hat{y}$ = sign($f(\mathbf{x})$), where $f(\mathbf{x})$ is a weighted sum of non linear basis functions (e.g., sigmoidal). In the case of RLS, generalization ability relies on two main concepts: the function $f(\mathbf{x})$ belongs to a Reproducing Kernel Hilbert Space (RKHS), and Regularization Theory is used as the conceptual basis [3]. The decision function $f_{\mathrm{RLS}}(\mathbf{x})$ can be written as:

$$f_{RLS}(\mathbf{x}) = \sum_{i=1}^{n} \beta_i K(\mathbf{x}, \mathbf{x_i}) \tag{1}$$

where, $K(,)$ is a kernel function [9] and $\boldsymbol{\beta} = (\beta_1, .., \beta_n)$ is a vector of scalar coefficients. It can be shown [9] that $\boldsymbol{\beta}$ is obtained by solving the following system of linear equations:

$$(\mathbf{K} + \lambda \mathbf{I})\boldsymbol{\beta} = \mathbf{y} \tag{2}$$

where, λ is a regularization parameter, and $\mathbf{K}$ is the matrix of kernel functions $K(\mathbf{s},\mathbf{x})$ that constitute the non linear basis functions [3]. In this work the well known RBF kernel is used [3]. The kernel width parameter σ^2 is set by a model selection procedure that is explained later.

SVM still uses kernel functions, $K(\mathbf{x}_i, \mathbf{x}_j)$. The decision function f_{SVM} is given by Eq. ($3):

$$f_{SVM}(\mathbf{x}) = \sum_{i}^{n_{sv}} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{3}$$

where, the number of Support Vectors n_{sv} , the '*bias*' term b and coefficients α_i are computed by the training algorithm [3], which minimizes a quadratic cost function [3].

Finally, RELM is a single-layer feedforward network that connects the input layer to the hidden layer (having N_h neurons) through a set of weights. As a result, the overall decision function, $f_{\mathrm{RELM}}(\mathbf{x})$ is given by:

$$f_{RELM}(\mathbf{x}) = \sum_{j=1}^{N_h} \hat{w}_j a_j(w_j \mathbf{x} + b_j) + \hat{b} \tag{4}$$

where, $a_j()$ is a nonlinear activation function, and the terms b_j w_j are randomly set. The training reduces to the minimization of the following cost function:

$$\min_{\{\hat{\mathbf{w}},\hat{b}\}} \left\| \mathbf{H}\,\hat{\mathbf{w}} - \mathbf{y} \right\|^2 + \lambda \left\| \hat{\mathbf{w}} \right\|^2 \tag{5}$$

where $\mathbf{H}$ is an "activation matrix", such that the entry $\{h_{ij} \in \mathbf{H}; i=1,...,N; j=1,...,N_h\}$ is the activation value of the j-th hidden neuron for the i-th input pattern. The quadratic loss term in (5) is augmented by the regularization term $\lambda \left\| \hat{\mathbf{w}} \right\|^2$.

B. Implementation in Embedded Electronic Systems

The above classification methods can be effectively implemented by an electronic embedded system for real-time prediction - thanks to the simple expression of the decision functions. This aspect makes them -in principle- preferable to tools such as Naïve Bayes or Decision Trees, which do not support efficient hardware implementations.

In practice, the eventual complexity of the decision function depends on the particular characteristics of each tool. In RLS the decision function complexity scales with the number of training samples n. In SVM, several α_i coefficients (as per Eq. (3)) are null at the end of the training process. Hence, its decision function implies some compression in the classification process, as eventually $n_{\mathrm{sv}} < n$ where n is the number of training samples. On the other hand, the computational complexity of the RELM decision function scales with the number of neurons in the hidden layer, N_h; that parameter is set in relation to prediction accuracy by exploiting practical criteria such as cross validation [3].

IV. Simulation Results

A. Simulation Setup

The dataset used for the simulations was generated following the procedure described in Sec. II without any preprocessing; i.e. raw data were used. The dataset $\boldsymbol{X}$ included n = 600 patterns, each pattern being composed by (m = 4): input frequency, input force, output phase, output force. Each of the four elements of the data pattern were normalized in the range [-1, +1] except for the multiclass RELM problem which achieved better performance without normalization.

To evaluate the performance of the CI paradigms under different configurations the simulation involved three tests:

- A: all the four features fed the classification systems;
- B: the classification systems received as input: the input frequency, the sensor output phase, and amplitude;
- C: the classification systems received as input only two features: the sensor output phase, and amplitude.

The problem setup C was particularly designed to assess the performance of the proposed framework under a configuration that only involved the quantities generated as output of the tactile sensing chip.

A cross-validation strategy [3] supported all the

TABLE I
COMPARISONS AMONG CLASSIFIERS: 2-CLASS, PROBLEM SETUP: A

Problem	Validation Errors %			Neurons/S.V.	
	RELM	RLS	SVM	RELM	SVM
Brass VS Wood	**7%**	9%	8%	95	**45**
Brass VS Plastic	**0%**	2%	1%	**35**	68
Brass VS Lead	**7%**	**7%**	8%	**65**	76
Wood VS Plastic	**0%**	**0%**	2%	**15**	70
Wood VS Lead	**0%**	**0%**	0%	**15**	90
Plastic VS Lead	**0%**	**0%**	0%	**25**	91
Average	**2.3%**	3.0%	3.2%	**41.7**	73.3

TABLE II
COMPARISONS AMONG CLASSIFIERS: 2-CLASS, PROBLEM SETUP: B

Problem	Validation Errors %			Neurons/S.V.	
	RELM	RLS	SVM	RELM	SVM
Brass VS Wood	15%	9%	**8%**	85	**47**
Brass VS Plastic	**0%**	2%	1%	**25**	68
Brass VS Lead	10%	**8%**	10%	**25**	56
Wood VS Plastic	1%	**0%**	2%	**35**	70
Wood VS Lead	2%	**0%**	**0%**	**25**	88
Plastic VS Lead	2%	**0%**	**0%**	**15**	89
Average	5.0%	**3.2%**	3.5%	**35.0**	69.7

TABLE III
COMPARISONS AMONG CLASSIFIERS: 2-CLASS, PROBLEM SETUP: C

Problem	Validation Errors %			Neurons/S.V.	
	RELM	RLS	SVM	RELM	SVM
Brass VS Wood	21%	**17%**	18%	135	**37**
Brass VS Plastic	8%	**2%**	3%	**25**	32
Brass VS Lead	12%	5%	**4%**	**15**	31
Wood VS Plastic	**3%**	4%	4%	**55**	65
Wood VS Lead	4%	**0%**	**0%**	**15**	89
Plastic VS Lead	7%	**0%**	**0%**	**45**	69
Average	9.2%	**4.7%**	4.8%	48.3	53.8

experiments: the dataset was randomly split into a training set, a validation set, and a test set. In the training phase, the validation set supported model selection, i.e. the selection of the most effective parameterization for the CI tool. Eventually, the selected model was assessed by measuring the accuracy on the test set, thus simulating a “on the field” performance.

The experimental campaign dealt with the multi-class problem characterizing the dataset and all the bi-class problems that could be derived from it. Bi-class simulations allowed a detailed analysis of the complexities of the different binary problems involved. The multiclass simulation assessed the ability of the proposed approach to tackle the applicative domain. In the bi-class problems, the training, validation and the test sets included 150, 50 and 100 patterns, respectively. The following parameters were used for model selection:

- RELM: $N_h \in [5, 200]$ by steps of 10 neurons; $\lambda = \{2^{-16}, 2^{-15},\dots, 2^{+15}\}$.
- RLS: RBF kernel with $\sigma = \{2^{-12}, 2^{-11},\dots, 2^{+12}\}$; $\lambda = \{2^{-12}, 2^{-11},\dots, 2^{0}\}$.
- SVM: RBF kernel $\sigma = \{2^{-12}, 2^{-11},.., 2^{+12}\}$; $C = \{2^{0},.., 2^{11}, 2^{12}\}$.

In the multiclass problem, the size of training, validation and test sets was of 300, 100 and 200 samples, respectively. The following parameters were used for model selection:

- RELM: $N_h\in[5,300]$ by steps of 5 neurons; $\lambda = \{2^{-12}, 2^{-11},\dots, 2^{0}\}$.
- RLS: RBF kernel with $\sigma = \{2^{-12}, 2^{-11},.., 2^{+6}\}$; $\lambda = \{2^{-12}, 2^{-11},.., 2^{-3}\}$.
- SVM: RBF kernel with $\sigma = \{2^{-12}, 2^{-11},.., 2^{+6}\}$; $C = \{2^{0},.., 2^{11}, 2^{12}\}$.

All simulations were run in Matlab. The ‘/’ Matlab operator was used to solve the linear systems involved in the training stage of RLS and RELM; for SVM learning, a C .mex file implementing the Sequential Minimal Optimization algorithm [16] was adopted.

B. *Simulation Results*

Tables I, II, III show the simulation results for the problem setup A, B, and C, respectively. In each table, the first three columns give the classification error on the test set obtained by RELM, RLS and SVM. The last two columns show the number of neurons used by RELM and the number of support vectors in SVM, respectively. In this model, RLS is not included as the number of computational units is always equal to the number of training patterns, *n*.

The results presented in Table I show that all the classifiers achieved a high accuracy level. SVM and RELM provided the best trade-off between accuracy and number of computational units. Table II confirms such evaluation. RLS and SVM achieved satisfactory performance, while the accuracy of RELM on the problem “Brass vs Wood” was fairly low. The results reported in Table III confirm that the problem “Brass vs Wood” was challenging. As expected, though, the classification performance of the three classifiers in general slightly worsened on the problem setup ‘C’. Overall, these results show that the proposed methods effectively coped with the current data classification domain.

The second set of simulations session dealt with multiclass classification. Alike the two-class experimental session, three different problem set-ups were tackled. To attain a robust estimation of the systems generalization performance, each experiment was repeated five times; each run involved a different composition of the training, validation, and test sets.

Table IV, V, and VI report the results. The mean and the standard deviation values of the classification errors are given for each classifier. Simulations confirm that SVM attained the best trade-off between accuracy and computational complexity of the eventual model implementation.

V. Discussion and Conclusions

This paper utilizes machine learning classifiers to classify the contact materials from raw tactile sensing data. Simulation results proved that CI paradigms can suitably tackle the material classification problem under different configurations of the input space i.e., different sets of features feeding the CI-based classification system. The robustness and reliability of the proposed approach are indeed confirmed by the results obtained in the multiclass problems, which involved five different, independent simulation runs. This is an important outcome of the research, which confirms that CI paradigms could effectively address tactile sensing.

A thorough comparison with other CI-based approaches to tactile data processing and classification [7, 8] is difficult because of the lack of a common testbed. In [7], surface texture pattern identification was addressed. The paper compared the performance of SVM, Decision Tree, and Naive Bayes. Five different surfaces were involved in the experiments. The paper shows that the best results in terms of classification accuracy were achieved by SVM or Decision Tree. In [8], a Naive Bayes classifier was exploited to distinguish textures sensed by a bio-inspired artificial finger. The experimental session involved seven different materials. Reported results give 83.5% as best classification accuracy.

The comparison between the experimental results presented in this paper and the previous works seems to confirm that the proposed classification paradigms can improve the classification accuracy. Such outcome indirectly proves the ability of the three paradigms to tackle challenging problems without any pre-processing of the input features i.e. simply using raw tactile input data. In this regard, one should also take into account that SVM and RELM are hardware friendly, while approaches based on Naive Bayes classifiers or Decision Trees do not exhibit such feature.

Future works will extend the proposed approach to classify materials into more classes. Increasing the number of classes (i.e., materials in our case) may affect the overall performance in terms of classification accuracy; nonetheless satisfactory results (classification accuracy > 80%) can be obtained when the classes are not highly overlapped in the input space. When this condition does not hold, any classification paradigm would fail in supporting a reliable multiclass system [3]. Another dimension of future activity concerns implementing the classification tool on embedded systems and hence support the hardware implementation of the complete tactile sensing system.

TABLE IV
Comparisons Among Classifiers: Multiclass, Problem Setup: A

Measure	RELM	RLS	SVM
Global Error %	11.4%(3.54%)	**10.5%(1.87%)**	11.2%(1.39%)
Neurons or S.V.	**184(57.8)**	1200	258.4(28.68)

TABLE V
Comparisons Among Classifiers: Multiclass, Problem Setup: B

Measure	RELM	RLS	SVM
Global Error %	12.3%(4.57%)	**11.6%(3.99%)**	12.6%(3.86%)
Neurons or S.V.	**124(60.2)**	1200	187.2(27.0)

TABLE VI
Comparisons Among Classifiers: Multiclass, Problem Setup: C

Measure	RELM	RLS	SVM
Global Error %	37.8%(3.1%)	21.6%(2.94%)	**20.8%(2.88%)**
Neurons or S.V.	**26(28.15)**	1200	287.4(32.3)

Acknowledgments

The research leading to these results has partly received funding from European Community's 7th framework "People" specific programme – Marie Curie Actions - under grant agreement PCOFUND–GA–2008 – 226070, Trentino Project.

References

[1] R.S. Dahiya, G Metta, M. Valle, G. Sandini, "Tactile Sensing—From Humans to Humanoids," *IEEE Trans. on Robotics*, vol. 26, no. 1, 2010, pp. 1-20

[2] R. S. Dahiya, G. Metta, M. Valle, A. Adami, and L. Lorenzelli, "Piezoelectric oxide semiconductor field effect transistor touch sensing devices," *Applied Physics Letters*, vol. 95, pp. 034105-034105-3, 2009.

[3] B. Schölkopf and A. J. Smola, *Learning with kernels*, MIT Press, December 2001.

[4] C. O. Jaynes, "Seeing is Believing: Computer Vision and Artificial Intelligence," *Crossroads* – Special Issue on artificial Intelligence, vol. 3, no. 1, 1996, pp 7-10.

[5] M. Talbot, M. Arvandi, A. Sadeghian, "A Neural Network Based Surface Roughness Discrimination Algorithm," *Proc. of the Int. Symp. on Intelligent Automation and Control*, Gold Coast, Australia, 2008, pp. 1-8

[6] A. J. Worth and R.R. Spencer "A Neural Network for Tactile Sensing: The Hertzian Contact Problem" *IEEE Trans. on Systems, Man. and Cybernetics*, vol. 22, No. 1, 1992, pp. 177-182

[7] A. M. Mazid, A. B. M. Shawkat Ali, "Opto-tactile Sensor for Surface Texture Pattern Identification using Support Vector Machine", 2008 *Proc. 10th Intl. Conf. on Control, Automation, Robotics and Vision*, Hanoi, Vietnam, 17–20 December 2008, pp. 1830 - 1835

[8] N. Jamali and C. Sammut, "Material Classification by Tactile Sensing using Surface Textures," *Proc. IEEE Int. Conf. on Robotics and Automation*, Anchorage, USA, 2010, pp. 2336 - 2341

[9] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization Networks and Support Vector Machines," *Journal Advances in Computational Mathematics*, Springer, vol. 13, no. 1, 2000, pp. 1-50.

[10] G.B. Huang, Q.Y. Zhu, C.K. Siew, "Extreme learning machine: Theory and applications", *Neurocomputing*, Vol. 70 pp. 489–501, 2006.

[11] R. S. Dahiya, G. Metta M. Valle, "Development of Fingertip Tactile Sensing Chips for Humanoid Robots," *Proc. 2009 IEEE Int. Conf. on Mechatronics*, Malaga, Spain, 2009, pp 1-6.

[12] R. S. Dahiya, M. Valle, G. Metta, and L. Lorenzelli, "POSFET Based Tactile Sensor Arrays," in *IEEE ICECS'07, The 14th Int. Conf. on Electronics, Ckts and Systems*, 2007, pp. 1075-1078.

[13] M. Redwood, "Transient Performance of Piezoelectric Transducers," *The J. of the Acoustical Society of America*, Vol. 33, pp. 527-36, 1961.

[14] R. Dahiya, M. Valle, and L. Lorenzelli, "SPICE model for lossy piezoelectric polymers," IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control, vol. 56, pp. 387-395, 2009.

[15] R. Rifkin, A. Klautau, "In Defense of One-Vs-All Classification", *Journal of Machine Learning Research*, Vol.5, pp. 101-141, 2004.

[16] C.C. Chang, C.J. Lin, "LIBSVM: A Library for Support Vector Machines", 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm