Support Vector Machine Learning for Gesture Signal Estimation with a Piezo-Resistive Fabric Touch Surface

Andrew Schmeder
Center for New Music and Audio Technologies
University of California, Berkeley
1750 Arch Street
Berkeley, CA, 94720
andy@cnmat.berkeley.edu

Adrian Freed
Center for New Music and Audio Technologies
University of California, Berkeley
1750 Arch Street
Berkeley, CA, 94720
adrian@cnmat.berkeley.edu

ABSTRACT

The design of an unusually simple fabric-based touch location and pressure sensor is introduced. An analysis of the raw sensor data is shown to have significant non-linearities and non-uniform noise. Using support vector machine learning and a state-dependent adaptive filter it is demonstrated that these problems can be overcome. The method is evaluated quantitatively using a statistical estimate of the instantaneous rate of information transfer. The SVM regression alone is shown to improve the gesture signal information rate by up to 20% with zero added latency, and in combination with filtering by 40% subject to a constant latency bound of 10 milliseconds.

Keywords

gesture signal processing, touch sensor, support vector machine

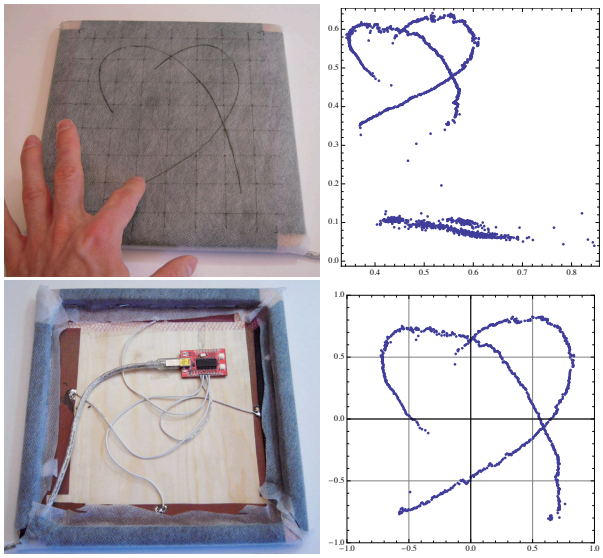

Figure 1: Clockwise from top left: Prototype fabric touch and pressure sensing surface. Raw sensor data. Gesture estimation by SVM. Electronics.

1. INTRODUCTION

Recent advances in conductive polymers have enabled a new generation of electronic circuits and sensors using flexible materials [5]. In this paper a prototype touch surface sensing a single spatial location and applied pressure is constructed using piezo-resistive fabric (Figure 1). An advantage of piezo-resistive fabric over other materials such as rubber is its fast recovery from compression, enabling high-temporal resolution as required for intimate control of an electronic musical instrument. The unusually simple design shown can be constructed in less than 10 minutes using basic tools. However, while these malleable materials enable extremely rapid design and prototyping, the sensing strategies are still experimental and have difficulties including non-linear coupling between measurements and high-noise gain at the edge of the valid sensing area. These problems can be solved without significantly increasing the latency of the sensing by using machine learning and adaptive filtering. The theory, implementation and evaluation of this method is described in this paper.

2. THE FABRIC TOUCH SURFACE

The design of a surface for simultaneous sensing of pressure and touch location can be realized with two sheets of piezo-resistive fabric separated by a standoff layer of light-weight plastic mesh. The tools required are: scissors to cut the materials, soldering iron for wiring to a microcontroller, and a staple gun to fix the fabric to a frame. The design is shown in schematic form in Figure 2 and a prototype example is photographed in Figure 1.

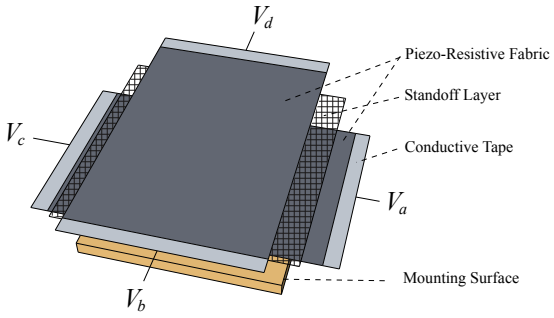

Figure 2: Design schematic for the touch and pressure sensing surface. Sheet materials are folded over and stapled to mounting surface.

The design has many variables including the type of fabric, standoff layer material, size of the surface, and use of constant current or constant voltage sensing. The prototype shown here is approximately 20cm^2 constructed

with EeonTex (tm) piezo-resistive "switch" fabric [3], a specially coated fabric with low resistance along the surface of one side and a higher resistance through the z-axis of the material.

2.1 Equivalent Circuit

An ideal electric circuit for the touch surface is shown in Figure 3.

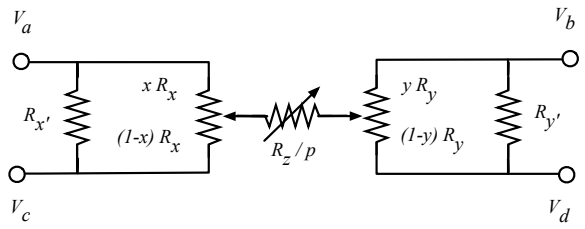

Figure 3: Ideal circuit model of touch pressure sensor

The nodes (V_a, V_b, V_c, V_d) are the wires shown in Figure 2, and are connected directly to the pins of a microcontroller with bi-directional drive (input or output) and analog voltage measurement capability. The position of the tap on resistors R_x and R_y are determined by the (x, y) coordinate of a single object touching the surface, and the pressure p applied to R_z gives a variable resistance at the point of contact. The standoff-layer ensures that $p \to 0$ when no pressure is applied by physically pushing the layers apart. The additional terms $R_{x'}$ and $R_{y'}$ represent unknown (presumably small) cross-talk components induced by the radial spread of current around the point of touch.

Provided that the cross-talk components $R_{(x',y')}$ are small and that that the transverse resistance $R_{x,y}$ is typically at least one order of magnitude smaller than the through-resistance R_z/p (e.g., by application of a surface coating to achieve higher conductivity parallel to the surface), then the measurements of each parameter will be nearly-linear and independent (see also Section 3.5.2 in [7] for more details). In reality this ideal is best approximated only at the center of the sensing surface.

2.2 Sensor Measurement Method

To obtain a measurement of the parameters (x, y, p), a program on the microcontroller is used to drive the nodes in a sequence of four states listed in Table 1 to obtain eight voltages. At each step, two of the nodes are output states comprising a source and a sink causing current to flow along a path through the fabric that is essentially oriented along the x, y or p direction, while two other nodes are input states measuring the voltage. The resulting 8 channels of measurement are sampled at a rate of 1000hz and the skew between measurements is less than 100 microseconds.

	V_a	V_b	V_c	V_d
Step 1:	x	1	$(1-x)$	0
Step 2:	1	y	0	$(1-y)$
Step 3:	1	0	$(1/p)$	$(1/p)$
Step 4:	$(1/p)$	$(1/p)$	1	0

Table 1: Four-step measurement sequence. A state of 1 or 0 indicates the pin in output mode driven high or low.

2.3 Parametric Sampling of Raw Sensor Data

Registration marks were drawn on the top of the touch surface prototype with a pen at 2.5cm intervals yielding an 8x7 grid. A calibration jig was constructed that could be adjusted to apply a controlled pressure at each grid point with a spatial distribution approximately the size of a finger tip. Stimulation of the surface at each point with 5 different levels of pressure was performed by changing the weight of the jig on a $\log_2$ scale and with $p = 0$ corresponding to the smallest weight producing sufficient pressure to overcome the standoff barrier layer. The eight measurement voltages were recorded for each of the 8x7x5 = 280 conditions and labelled with the corresponding (x, y, p). This dataset is representation by the notation:

$$\mathbf{D} = (\mathbf{X}, \mathbf{Y}) = \{(\mathbf{x}_i, \mathbf{y}_i) : i = 1 \ldots 280\} \quad (1)$$

Where $\mathbf{X} \in \mathbf{R}^{8\times 280}$ is matrix of samples where each column is one of the eight channels of raw sensor data, and $\mathbf{Y} \in \mathbf{R}^{3\times 280}$ is the matrix of the the corresponding parameters of each sample (x_i, y_i, p_i).

2.4 Evidence of Non-Linearity

Three channels of $\mathbf{X}$ corresponding to measurements of (x, y, p) as described in Section 2.2 were put into a matrix and compared with a correlation analysis. These columns will be notated $\mathbf{x}_{(x,y,p)}$. The resulting correlation matrix is:

	$\mathbf{x}_x$	$\mathbf{x}_y$	$\mathbf{x}_p$
$\mathbf{x}_x$	1.	0.04	0.47
$\mathbf{x}_y$	0.04	1.	0.15
$\mathbf{x}_p$	0.47	0.15	1.

This demonstrates relatively small correlation between $\mathbf{x}_{(x,y)}$ but a strong correlation between $\mathbf{x}_p$ and $\mathbf{x}_{(x,y)}$.

Given that the sampling of (x, y, p) in $\mathbf{Y}$ are linearly independent, each should account for the total variance in $\mathbf{X}$ in proportions equal to the sampling density (8,7,5). However, Principal Components Analysis applied to $\mathbf{X}$ found clear evidence that the data does not fit into a 3-dimensional subspace (Table 2).

Parameters	x	y	p
Number of Samples	8	7	5
Explanation of Variance Expected	40%	35%	25%
Variance in $PCA(\mathbf{X})_{(1,2,3)}$	36%	30%	27%
Residual Variance $PCA(\mathbf{X})_{(4\ldots8)}$	5%		
Residual Covariance in $PCA(\mathbf{X})$	$< 3\%$		

Table 2: Explanation of variance expected in D versus measured variance assuming linearity

In summary, the raw sensor data is correlated but cannot be decorrelated using a linear transform.

2.5 Other Issues in Raw Data Quality

2.5.1 Pressure Range Issues

When $p \to 0$ the resistance between layers goes to infinity. This condition occurs when a touch is released from the surface or when a touch is very light. As a result the spatial measurement of (x, y) becomes either impossible or, at the edge of the sensing range, is very noisy. A pressure-dependent adaptive filter is introduced later to deal with the state-dependent variation in noise and to extend the useful range of measurement.

2.5.2 Material Imperfections

During the process of the systematic acquisition of $\mathbf{D}$ we noticed a significant deviation from the normal range of raw

data in a particular region of the surface. The abnormal area was limited to an area of only a couple centimeters diameter. Later we were informed by the supplier that the fabric coating may have local irregularities.

3. PARAMETER ESTIMATION BY SVR

The support vector machine (SVM) is an algorithm for making a predictive model that estimates a relationship between raw data and a discrete class label or continuous parameter. SVMs are well-known for their relatively simple controls, bounded computational cost, and direct relationship between the trained machine and the source training data through the retention of a sparse collection of examples called support vectors [4].

The epsilon-insensitive Support Vector Regression (ϵ-SVR) algorithm takes as input a training dataset

$$\{(\mathbf{x}_i, f(\mathbf{x}_i)), i = 1 \ldots \ell\}, \mathbf{x} \in \mathbf{R}^d, f(\mathbf{x}) \in \mathbf{R} \quad (2)$$

and produces an approximation of the real-valued continuous function f by optimizing the choice of (α_i, b) in:

$$f(\mathbf{x}) \approx \sum_{i=1}^{\ell} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \quad (3)$$

The α_i are constrained to lie in the interval $[-1, 1]$. If $\alpha_i \neq 0$ then the corresponding $\mathbf{x}_i$ is a support vector. If $|\alpha_i| = 1$ then it is a bounded support vector. Since only $\mathbf{x}_i$ with non-zero α_i are necessary in the reconstruction of f, the resulting SVM model is a sparse subset of the training data.

The kernel function K is chosen in this work to be the radial basis type (a typical choice for real-valued data):

$$K(\mathbf{x}, \mathbf{y})_{(\gamma,\theta)} = e^{-\gamma \sum_{i=1}^{d} \theta_i (\mathbf{x}_i - \mathbf{y}_i)^2} \quad (4)$$

The *kernel trick* is that $K(\mathbf{x}, \mathbf{y}) =< \Phi(\mathbf{x}), \Phi(\mathbf{y}) >$, where $\Phi(\mathbf{x}) : \mathbf{X} \rightarrow \mathbf{F}$ for $\mathbf{F}$ a very high (possibly infinite) dimensional feature space with dot product $< \cdot, \cdot >$. SVM performs linear separation in $\mathbf{F}$ using dual-optimization with the implicit map Φ (Figure 4).

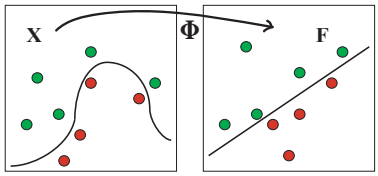

Figure 4: SVM solves a linear problem by means of an implicit mapping to F

3.1 SVR Model Optimization

The ϵ-SVR is controlled by the following variables:

- ϵ, the interval inside of which there is no penalty for regression error
- γ, the scaling factor of the radial basis kernel function
- $\theta = \{\theta_i : i = 1 \ldots d, \sum w_i = d\}$ normalized weight of each raw feature [2]

Prior to training the SVR, each raw feature is scaled independently to range over the interval $[-1, 1]$. The optimal selection of parameters $(\epsilon, \gamma, \theta)$ was performed by multi-resolution grid search minimizing the mean-squared error of 4-way cross-validation subject to bounds on the maximum number of support vectors (nSV) and minimum and maximum number of bounded support vectors (nBSV). The upper bound on nSV controls computational cost of the model. The lower and upper bounds on nBSV control under-fitting and over-fitting, respectively, which were chosen by manual adjustment to produce a reasonable result. The optimization objective is:

$$\min_{\epsilon,\gamma,\theta} \frac{1}{\ell} \sum_{i}^{\ell} (\epsilon\text{-SVR}_{(\epsilon,\gamma,\theta,\mathbf{D})}(\mathbf{x}) - f(\mathbf{x}))^2, \quad (5)$$

subject to

$$0 < \text{nSV} < \frac{2}{3}\ell,$$

$$\frac{1}{12}\text{nSV} < \text{nBSV} < \frac{1}{4}\text{nSV}.$$

Because ϵ-SVR estimates only a 1-dimensional function, this process was repeated for each of (f_x, f_y, f_p) resulting in three models. The geometry of the search space for ϵ and γ is shown in Figure 5. The final model results for each parameter are summarized in Table 3.

	x	y	p
Parameter ϵ	0.036	0.051	0.14
Parameter γ	0.52	0.58	1.20
nSV	62	56	76
nBSV	37	15	26
Mean Squared Error	0.0018	0.0030	0.021

Table 3: SVR model optimization results

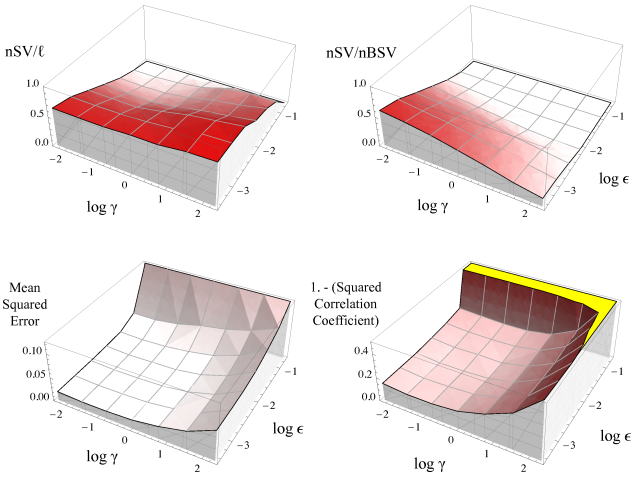

Figure 5: SVR parameter optimization space

3.1.1 *Concentration of SV Information*

The relationship between α_i and the spatial coordinates (x, y) are shown at each pressure level in Figure 6. As expected we observe a larger amplitude in α near the corners where the non-linear behavior of the circuit is largest.

3.1.2 *Implementation*

The implementation of SVM used was LIBSVM 2.9 [1] with modifications to support the θ parameter and an interface to the *Mathematica* notebook environment, which was the main tool for performing the multi-resolution grid optimization and other analyses.

4. STATE-DEPENDENT NOISE FILTERING

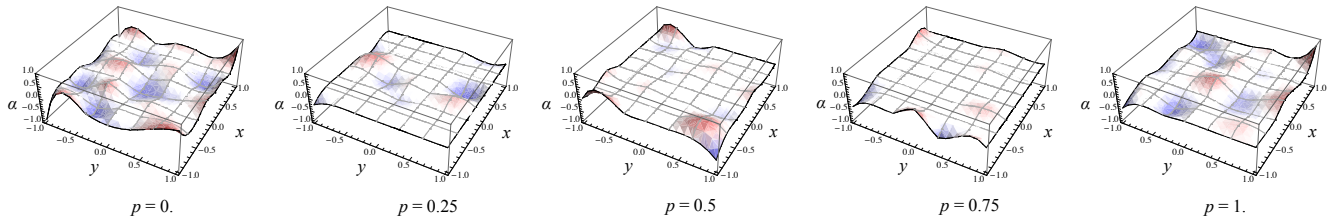

Figure 6: Magnitude of SVR coefficients α_i as a function of (x, y, p). Note strong deflections at corners.

4.1 Pressure-Dependent Spatial Noise

The increase in noise as $p \to 0$ has been touched on briefly in Section 2.5.1. A first approach to this problem is to place a thresholding on p below which the data is not considered valid. However this does not compensate for the noise still present in the valid sensing range. The standard deviation of this noise is state-dependent and approximately inversely proportional to p.

To measure the noise present after SVR remapping, a recording of typical touches at different pressure levels at each grid point was made, and from this the noise level with respect to the vector norm of spatial coordinates (x, y) estimated to be 0.12 at $p = 0$ and 0.06 at $p = 1$. Our goal was to make this noise uniform across all values of p subject to a bound on the delay incurred by the noise-removal filter.

4.2 Constant-Error Bounded-Delay Filtering

Assuming a normally-distributed Gaussian noise process with standard deviation σ, the standard error of the sample mean averaging over n samples is:

$$\sigma/\sqrt{n}. \tag{6}$$

For a time-varying signal where the instantaneous standard deviation is known (for example by measurement), then it is possible to formulate a filter that has a constant standard error by dynamically changing the size of the averaging window n. The total delay of the filter can be bounded by constraining n to be less than some limit.

Allowing for applications of the touch surface in computer-based musical instrument control, we choose the latency bound to be 10 milliseconds. At the sampling rate of 1000hz this is equivalent to a delay of 10 samples, i.e., the effective delay of the filter must not exceed $n = 10$. Given the maximum standard deviation σ' (e.g., the noise observed when $p \to 0$), the effective n to achieve constant standard error at σ subject to the delay bound is:

$$\frac{\sigma}{\sqrt{n}} = \frac{\sigma'}{\sqrt{10}}, \tag{7}$$

$$n = \frac{10\sigma^2}{\sigma'^2}. \tag{8}$$

Note that $\sigma < \sigma'$ and that $n < 10$.

Consider the discrete-time order-2 recursive low-pass filter acting on input f parameterized by ω,

$$g_{(f,\omega)}(t) = (\omega)g_{(f,\omega)}(t-1) + (1-\omega)f(t) \tag{9}$$

The effective delay n of this filter is the integral of its impulse response,

$$\int_0^\infty (\frac{1}{\omega})^{-t} dt = \frac{1}{\log \omega^{-1}}. \tag{10}$$

Therefore the appropriate selection of ω given a desired n is

$$\omega_n = e^{-\frac{1}{n}}. \tag{11}$$

Finally, a pressure-dependent filter for conditioning of $f_x(\mathbf{x}, t)$ and $f_y(\mathbf{x}, t)$ is given by

$$h_{(f_{(x,y)}(\mathbf{x}), f_p(\mathbf{x})(t))} = g_{(f_{(x,y)}(\mathbf{x}), \omega_{n_p})} \tag{12}$$

where ω_{n_p} is determined by estimation of the spatial standard deviation σ_p as a function of instantaneous pressure at time t (shown in Figure 7).

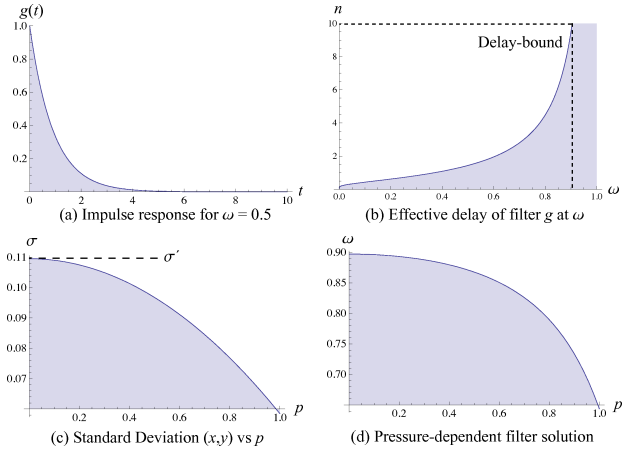

Figure 7: Derivation of the adaptive filter

5. EVALUATION

5.1 French Curve Test

Following the example data used in [7] two copies of a french-curve were drawn on the surface and recordings made of a finger stroke along the curves. The strokes were executed with normal dynamics and not controlled carefully for uniform pressure or velocity. Figure 8 shows the raw data, hard thresholding on pressure, noise removed by adaptive filtering, and final resulting data in (x, y, p) space.

5.2 Measuring Information Gain

From the graphs shown it is obvious that some improvement has been made, but a more quantitative analysis is needed. In the HCI community there exist task-oriented standards such as ISO-1941-9 for evaluation of pointing interfaces [6]. However for a broader applicability independent of any task, for example, in musical applications that are inherently task-free, this type of analysis is not useful.

Rather, we take the concept of the rate of information transfer, which is the primary determinant in Fitt's Law, and formulate an instantaneous version of the metric for a discretely-sampled stream at rate FS using the $\log_2$ ratio of the absolute-value of the moving average E_m of the derivative $d\mathbf{x}$ (i.e., "effective target distance") to 3 times the

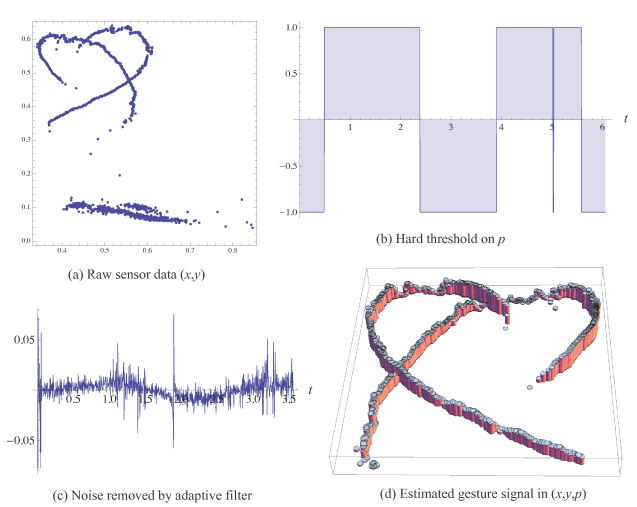

Figure 8: Processing french curves from raw to final

moving standard deviation (i.e., "effective target width").

$$I_m(\mathbf{x}) = \log_2 \frac{|E_m(d\mathbf{x})|}{3\sqrt{E_m((d\mathbf{x} - E_m(d\mathbf{x}))^2)}} \times \text{FS} \qquad (13)$$

Calculating I_m on the raw french curve data yielded a rate of information transfer ranging from 50 bits/sec to 90 bits/sec in the spatial (x, y) measurements. After estimation of the true parameters (x, y) using the ϵ-SVR models the information-rate increased by 10-20% with zero additional delay. After filtering with the latency-bounded filter the information-rate increased an additional 10-20% incurring a maximum delay of 10 milliseconds. Figure 9 shows the information-rate statistics across both french curves and Figure 10 shows the delay incurred by the adaptive filter.

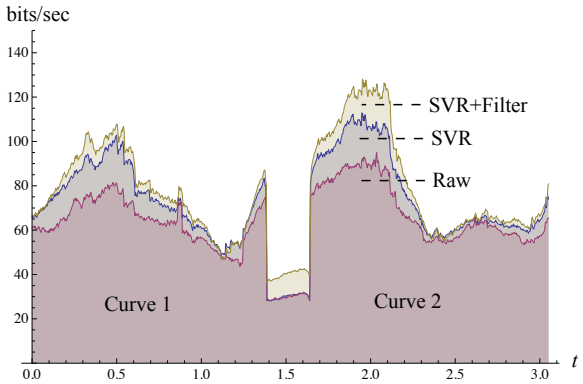

Figure 9: Spatial information rates over both curves

The mean and best-case improvement in estimation of (x, y, p) is shown in Table 4:

	Mean Improvement	Max Improvement
x	21.4%	43.5%
y	10.1%	20.03%
p	-0.2%	-2.9%

Table 4: Summary of information gain by parameter

Interestingly the gain in estimation of p was reduced compared to the raw data, which is not unsurprising considering the initial look at the raw data discovered excessive information coupled into that channel.

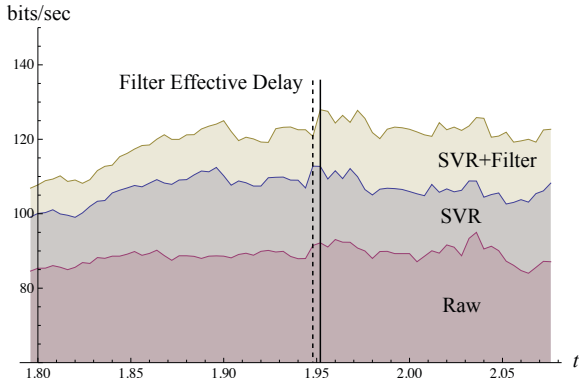

Figure 10: Detail of information transfer showing filter delay

Finally we look at the correlation matrix between predicted (x, y, p) from the ϵ-SVR and observe that the non-linear cross-talk between channels has been reduced to less than 2%.

	$f_x(\mathbf{x})$	$f_y(\mathbf{x})$	$f_p(\mathbf{x})$
$f_x(\mathbf{x})$	1.	0.00	0.02
$f_y(\mathbf{x})$	0.00	1.	0.01
$f_p(\mathbf{x})$	0.02	0.01	1.

6. CONCLUSION

6.1 Summary of Results

This paper shows that using a simple calibration rig, the raw data from fabric-based touch-pressure sensors can be processed to improve accuracy and remove non-linear cross-talk with a support vector machine learning approach. In addition the formulation of a constant-error bounded-delay filter was shown and used to gain information without compromising reactivity of the sensor as necessary in latency-sensitive applications such as music performance.

6.2 Future Work

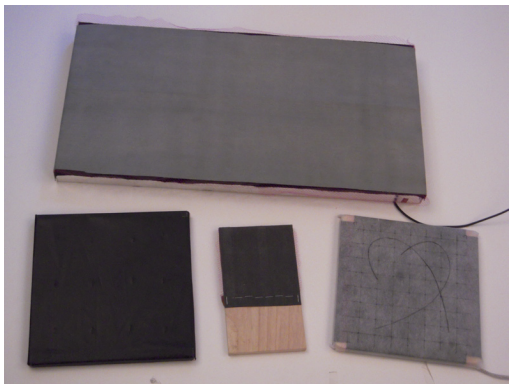

Figure 11: Variants on the basic design with different dimensions and materials

An interesting application of the method shown here, in particular the use of the moving rate of information transfer, can be used to provide quantitative answers to the impact of design decisions such as the size of the sensing surface and differences between constant-current and constant-voltage sensing. A collection of new surfaces was constructed that include some of these alternative choices, shown in Figure 11.

7. REFERENCES

[1] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[2] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.

[3] E. Corporation. EeonTex. `http://www.eeonyx.com/datasheets.php?type=eeontex`.

[4] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and Other Kernel-Based Learning Methods)*. Columbia University Press, 2000.

[5] A. Freed. Application of new fiber and malleable materials for agile development of augmented instruments and controllers. In *NIME 2008*, 2008. CNMAT.

[6] I. S. MacKenzie, T. Kauppinen, and M. Silfverberg. Accuracy measures for evaluating computer pointing devices. In *CHI '01: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 9–16, New York, NY, USA, 2001. ACM.

[7] I. Rosenberg and K. Perlin. The unmousepad: an interpolating multi-touch force-sensing input pad. In *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, pages 1–9, New York, NY, USA, 2009. ACM.