Program Digest, 2019 IEEE International Ultrasonics Symposium (IUS)
Glasgow, Scotland, October 6-9, 2019

Artificial Neural Networks for Impact Position Detection in Haptic Surfaces

Camilo Hernandez Mejia*, Paolo Germano*, Sebastian Correa Echeverri†, Yves Perriard, Senior Member, IEEE*
* *Laboratory of Integrated Actuators (LAI), École polytechnique fédérale de Lausanne EPFL*, Lausanne, Switzerland
camilo.hernandez@epfl.ch
† *Artificial Intelligence Engineer, Veridas*, Navarra, Spain
scorrea92@gmail.com

***Abstract*—Recently, the interest in haptic feedback is growing thanks to its ability to enhance the interaction with Human Machine Interfaces (HMIs). This research project is exploring the potential of machine learning combined with piezoelectric actuators to generate localized vibrational feedback over a thin rigid surface. With this goal in mind, this paper studied the potential of neural networks and machine learning algorithms to extract the position, where an impact has occurred. A data-set with 5310 stress signals labeled with the position at which the impact has occurred, was obtained using an automated Linear Impact Generator (LIG). Each signal was transformed into a spectrogram using the Fast Fourier Transform. During the study, different neural networks and machine learning algorithms were implemented and a supervised training process was carried out. At the end of the paper, the results of the different models are compared. The best model has an error (Validation MAE) of 4% and (Test MAE) of 8% in the impact position detection over an aluminum thin plate.**

***Index Terms*—Impact Position Detection, Surface Haptics, Piezoelectric Transducers, Machine Learning, Neural Networks.**

I. Introduction

In recent years, haptic feedback is gaining popularity since it can enrich the interaction with touch screens and other Human Machine Interfaces (HMIs). Moreover, haptics can enhance the usability of HMIs under extreme light/noise conditions where the traditionally used feedback methods (i.e. visual and auditive stimuli) are not available or are not as performant.

Preceding studies on the interaction of humans with acoustic musical instruments have shown that musicians perceive a rich haptic (i.e. Vibro-tactile) exchange with their instruments. This action-perception loop, not only influences the perceived quality of the instrument but, enhances the performance control (e.g. better timing or pitch control) while playing [1].

In the same context, Digital Musical Instruments (DMIs) commonly make use of touch screens as an interaction method, however, they generally lack haptic feedback. This is why restoring a proper haptic exchange between the DMI and the performer will dramatically improve the user experience. Furthermore, haptic enabled DMIs could give access to human beings with visual or hearing limitation to create music, by allowing them to feel the music.

Several approaches have been followed to create haptic feedback in tactile surfaces. For example, Time reversal method (TRM) has been studied in [4], [3], and [2] to focalize flexural waves in thin plates or cavities. Besides, in [8] TRM has also been used to detect the position were a finger impacts the surface. Alternatively, in [5] an array of 248 piezoelectric actuators was used to render localized haptic feedback within the area occupied by each transducer.

Project funded by the Swiss National Science Foundation (SNSF # 178972)

This research project is exploring the potential of machine learning and piezoelectric actuators to create converging elastic waves over a rigid surface, thus, obtaining multi-point and localized vibrational feedback.

Machine learning, specifically deep learning, has demonstrated promising results in the field of signal processing, even outperforming the classic signal processing methods [6]. In [7], the authors used data from finite element analysis (FEA) simulations and Artificial Neural Networks (ANN) to detect the position where an impact has occurred in the wing of an airplane that is made up with composite materials.

This paper continues to study the potential of neural networks and other machine learning algorithms for signal processing. In particular, a first approach to understand the ability of existing neural networks and machine learning algorithms to extract different features from the stress signals that can be measured at one or several locations of a thin rigid plate, after an impact occurs. Such signals are acquired by a single piezoelectric transducer bonded to a tactile surface (e.g. a beam of aluminum). Section II describes the procedure to acquire the signals and create the dataset. Then, section III presents the prior data analysis to identify the distribution of the data. Finally, section IV illustrates the training of the models that were trained and the different results.

II. Data Acquisition

To obtain the training dataset an automated linear impact generator (LIG) was used. An impact is induced in different positions over the surface of an aluminum beam, after each impact, flexural waves propagate on the surface diverging from the contact point, piezo-patches are used to convert the mechanical signal into an electric signal, the voltage across the electrodes of the piezoceramic patch is acquired using an oscilloscope.

A. Automated Linear Impact Generator

In [9] the development of a fully automated linear impact generator is described. A custom made electromagnetic linear actuator is mounted on a CNC table and an automation software was developed to consistently repeat the impact within an area of interest. The acquisition process is as follows: First, a matrix of impact positions is defined. Then, the LIG is moved to the first position on the surface that is being studied. Later on, the LIG induces a single impact and one or multiple piezoelectric patches and an acquisition system (in this case a LeCroy Waverunner LT224 oscilloscope) are used to record the voltage signals. The acquisition is repeated a defined number of times at the same position and then the LIG is moved to the next point. The complete system is presented in fig. 1

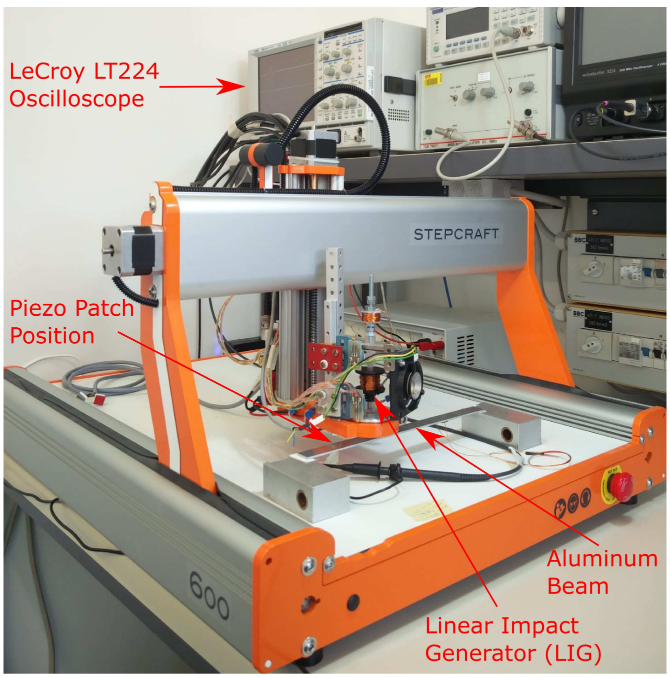

Fig. 1. Side view of the automated acquisition system that records the elastic waves in a fixed position of an aluminum beam after an impact is generated in multiple locations [9]

B. Acquisition Methodology

For this particular study, an aluminum beam of 250 mm x 16 mm x 2 mm (Length, Width, Thickness) as shown in fig. 2 was mounted on a CNC table using double-sided tape, a single piezoelectric transducer, located 62.5 mm far from the left border of the beam was used to acquire the signals. The study started at position X = 63 mm and finished at position X = 240 mm, since an impact was generated every 1 mm, 177 unique positions were studied. At every position, thirty (30) samples (i.e. single impact repetitions) where acquired. As a consequence, 5310 impacts were acquired in total. Note that for this experiment the Y position is kept constant during the whole study (Y = 8 mm is the midpoint of the beam).

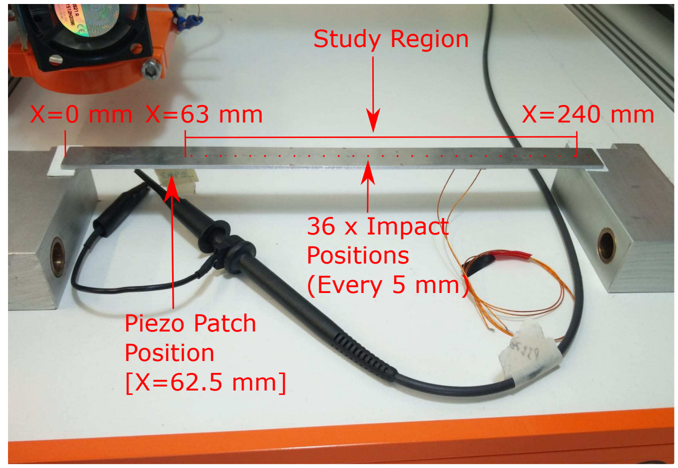

Fig. 2. An aluminum beam is mounted on the Stepcraft 600 CNC table using double sided tape. A piezoelectric patch is stuck under the beam at position x = 62.5 mm, and the study region goes from x = 63 mm and goes until 240 mm

For training purposes, (see section IV), the whole acquisition was divided into 5 different data-sets, each of them with a separation of 5 mm between consecutive impact positions. The conditions for the acquired datasets where as follows:

- Data-set # 1: Starting at X = 63 mm with steps of 5 mm until X = 238 mm (e.g. X = 63, X = 68, ... X = 238).
- Data-set # 2: Starting at X = 64 mm with steps of 5 mm until X = 239 mm (e.g. X = 64, X = 69, ... X = 239).
- Data-set # 3: Starting at X = 65 mm with steps of 5 mm until X = 240 mm (e.g. X = 65, X = 70, ... X = 240).
- Data-set # 4: Starting at X = 66 mm with steps of 5 mm until X = 236 mm (e.g. X = 66, X = 71, ... X = 236).
- Data-set # 5: Starting at X = 67 mm with steps of 5 mm until X = 236 mm (e.g. X = 67, X = 72, ... X = 237).

An example of the acquired signal is presented in fig. 3. To synchronize all the acquisitions of the dataset, the triggering signal of the LIG (i.e. a signal that triggers a single impact) was also acquired, this signal also triggers the acquisition on the oscilloscope.

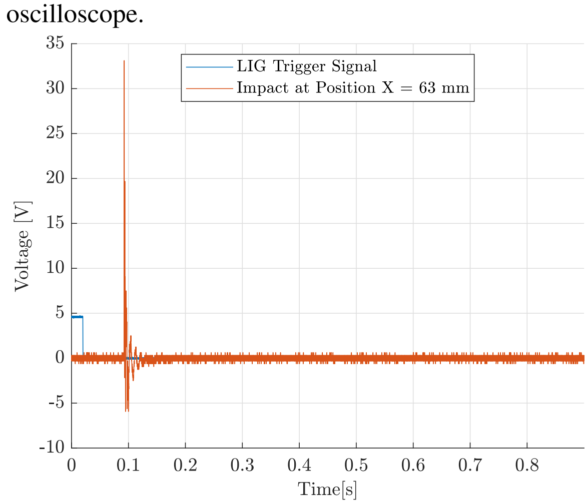

Fig. 3. Single acquisition after an impact was generated in the position X = 63 mm, the triggering signal starts the single impact

After every impact, the acquired voltage signals are transferred and stored in a computer for post-processing. The Fast Fourier Transform FFT function in Matlab is used to generate a spectrogram. A window of 256 points, an overlap of 64 points, a fast Fourier transform window of 256 points, and a sampling frequency of 50 kHz are defined. As a result, a matrix with 128 x 234 ”pixels” or data points is obtained. An example of graphic representation is displayed in fig. 4.

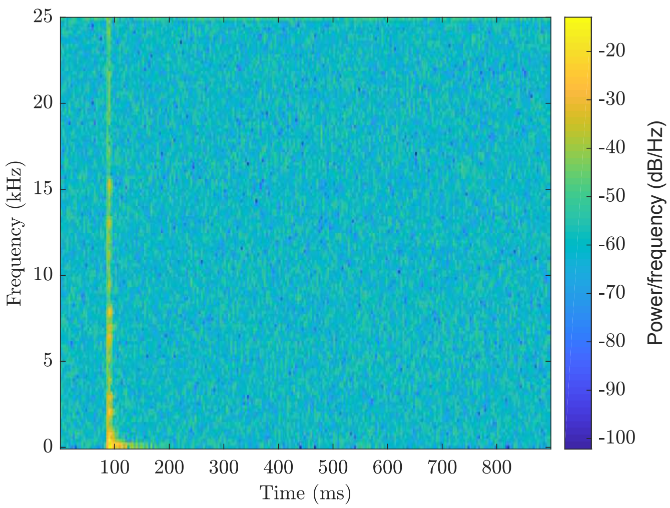

Fig. 4. A spectrogram is generated using the Fast Fourier Transform FFT function in Matlab, a window of 256 points is defined, with an overlap of 64 Points, an FFT window of 256 points, and the sampling frequency at the moment of the acquisition was 50 kHz

This matrix is reshaped into a single column vector with dimension 29952. After all acquisitions are done, a dataset is created by appending, side by side, all the spectrogram vectors and including a label with the position where the impact was generated and the number of the sample (e.g. ”acquisitionX63mm_Y8mm_sample_1”).

III. Data Analysis

Prior to training the neural networks and machine learning algorithms, the data was analyzed with the Principal Component Analysis (PCA) method, this statistical model allows to transform the data into a lower-dimensional version of the original data, the final set of orthogonal components (i.e. vectors) is a linear combination of the original data [10]. In other words, one can use PCA to find a lower-dimensional representation of the data, by scarifying precision, to understand the separability of the data and to obtain an intuitive visualization of the data distribution.

Scikit-learn python module [12], which implements the randomized truncated singular value decomposition (SVD) method by Halko *et al.* [11], is used to obtain the PCA in 2D and 3D (i.e. transformation to 2 and 3 main components). The results for PCA 3 show a promising data separation where the 30 samples at each point seem to agglomerate in a specific region presenting a high chance of obtaining classification or regression models, the results are presented in fig. 5.

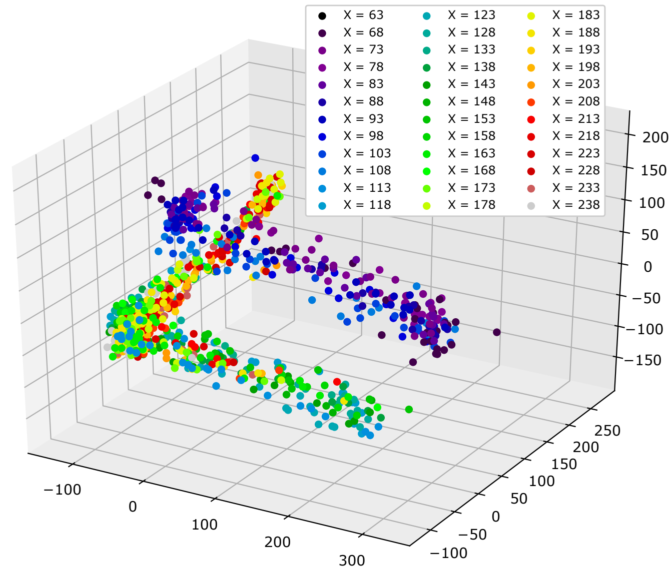

Fig. 5. PCA 3 for Dataset #1. Graphic representation of the reduction to the 3 principal vectors (i.e orthogonal components with the highest variance), each set of acquisitions at a specific position is represented by one unique color

IV. Models Training and Results

The impact position detection task was considered as a regression problem. In this manner, it is possible to obtain a continuous prediction of the impact position that allows the model to generalize. In other words, to detect the location of the impact even if input data comes from other positions that were not observed by the model during the training stage.

A. Data Preparation

Three data-sets (Data-set # 1, # 2, and # 3) were merged together, then, this data is randomly divided into 2 groups, 80% is used for training and 20 % is used for validation after each iteration of the learning process. Finally, the data-set # 4 and # 5 are used for testing the trained models, note that these two sets of data contain positions that have never been observed by the models.

B. ML Models and Neural Networks

Initially, two machine learning algorithms are fitted to predict the impact position, the linear regression model from Scikit-learn [12] and the XGB-Regressor from the XGBoost python module that is an implementation of the gradient boosted decision trees [13].

Later, the TensorFlow library [14] was used to implement two neural network models. First, a Recurrent Neural Network (RNN) [10] was implemented. This RNN has 4 hidden layers with ReLU activation, batch normalization and Gaussian noise (to improve the generalization). The output layer of the RNN has a linear activation function to help the model to behave as a regression. Second, a 2D Convolutional Neural Network (CNN) was implemented. The CNN has four convolutional layers, the first two layers with a depth of 32 and the second 2 layers have a depth of 64. These layers, have ReLU activation, MaxPooling to reduce the dimension of the input and a dropout of 0.5 that helps to improve the generalization of the model. The output of the convolutional layers is flattened and passed to a fully connected NN with one hidden layer with ReLU activation and an output layer with linear activation.

Both models use the mean absolute error (MAE) as the loss function and the Adam optimizer [15] was used instead of the classical stochastic gradient descent procedure to update the network weights based on training data. Last but not least, for the training of the networks a high learning rate (Lr = 0.001) was used during 100 epochs to help reduce the error, then a lower learning rate (Lr = 0.00001) was used through 900 epochs.

C. Results and Discussion

Table I presents the results obtained for the four models during the training (using the validation set of data) and during the test of the models (using the test data-sets # 4 and # 5). All the error values come from the MAE and the percentage of error is calculated considering the length of the working area 177 mm.

TABLE I

Error	LinearReg	XGBoost	NN	CNN 2D
Validation	19.5 / 11.2%	7.6 / 4.3%	14.9 / 8.4%	8.1 / 4.6%
Test # 4	22.2 / 12.6%	12.4 / 7.0%	21.6 / 12.2%	15.7 / 8.9%
Test # 5	31.7 / 17.9%	20.4 / 11.5%	26.9 / 15.2%	14.8 / 8.4%

The best models are the XGRegressor and the 2D CNN. The XGRegressor has the best results for the particular training dataset (merge of data #1, #2, and #3) but it does not have a good generalization when evaluated in the test data (Data # 4and # 5) that contain un-seen data points (i.e. contain a different population distribution). Similarly, the 2D CNN presents good results for the training data, however, it also presents coherent values for both test data-sets, which means that it generalizes in a precise manner. It is important to highlight that the 2D CNN presented an average MAE of 13 mm while the average validation MAE was 8 mm, this shows that the model still needs more exploitation and deepening. The fact that the training error continues above the validation error, means that the model needs more data to reduce the error while maintaining a high generalization.

V. Conclusions

In general terms, it was expected that the 2D CNN presents better results, as its approach uses kernels to find the 2D relationships of data. Owing to the fact that the signals are all synchronized it is possible for the network to find the relations between frequency and time for each spectrogram. It makes sense to continue studying this approach with more data (e.g. different materials, additional positions, among other variations). In addition, future studies should consider other neural networks with deeper architectures.

The obtained results will be extended to the extraction of the 2 dimensional (2D) position of the impact. Then, the analysis of impact signals acquired from multiple piezoelectric transducers. Finally, further analysis will be carried out in different materials to study the generality of the obtained models.

In conclusion, this study helps to understand the behavior and performance of machine learning for impact position detection in the particular domains that will be used for haptic feedback (i.e. thin rigid plates). Also, provide an alternative method for detecting the position of the finger in the tactile surface of the DMI.

References

[1] T. S. Papetti and C. Saitis, ”Musical Haptics”. Springer International Publishing, 2018.
[2] C. Hudin, J. Lozada, and V. Hayward, “Localized Tactile Feedback on a Transparent Surface through Time-Reversal Wave Focusing”, IEEE Transactions on Haptics, vol. 8, 2015.
[3] X. Liu, Y. Civet, and Y. Perriard, “Piezoelectric tactile device feedback generator using acoustic time reversal method”, in 2016 19th International Conference on Electrical Machines and Systems (ICEMS), pp. 1–5, 2016.
[4] S. Wöckel, U. Steinmann, and H. Arndt, “Haptics by time reversal of elastic waves”, in 2016 IEEE International Ultrasonics Symposium (IUS), 2016.
[5] S. Papetti, S. Schiesser, and M. Fröhlich, “Multi-Point Vibrotactile Feedback for an Expressive Musical Interface,” in Proceedings of the International Conference on New Interfaces for Musical Expression, pp. 235–240, 2015.
[6] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang, and T. Sainath, “Deep Learning for Audio Signal Processing”, IEEE J. Sel. Top. Signal Process., pp. 206–219, 2019.
[7] M. Ghajari, Z. Sharif-Khodaei, and F. M. H.Aliabadi, “Impact Detection Using Artificial Neural Networks”, Key Engineering Materials, 2012.
[8] M. R. Bai and Y. K. Tsai, “Impact localization combined with haptic feedback for touch panel applications based on the time-reversal approach”, The Journal of the Acoustical Society of America, vol. 129, pp. 1297–1305, 2011.
[9] C. Hernandez, J. Jayet, P. Germano, A. Thabuis, and Y. Perriard, ”Linear Impact Generator for Automated Dataset Acquisition of Elastic Waves in Haptic Surfaces”, in 2019 22nd International Conference on Electrical Machines and Systems (ICEMS), 2019.
[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016.
[11] N. Halko, P.-G. Martinsson, and J. A. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions”, arXiv:0909.4061 [math], 2009.
[12] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python”, J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
[13] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System”, in Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.
[14] M. Abadi et al., “TensorFlow: A System for Large-scale Machine Learning”, in Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, pp. 265–283, 2016.
[15] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization”, 2014.