2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Improved Normal and Shear Tactile Force Sensor Performance via Least Squares Artificial Neural Network (LSANN)

Meng Yee (Michael) Chuah[1,2] and Sangbae Kim[1]

Abstract— **This paper presents a new approach to the characterization of tactile array sensors that aims to reduce the computational time needed for convergence to obtain a useful estimator for normal and shear forces. This is achieved by breaking up the sensor characterization into two parts: a linear regression portion using multivariate least squares regression, and a nonlinear regression portion using a neural network as a multi-input, multi-output function approximator. This procedure has been termed Least Squares Artificial Neural Network (LSANN). By applying LSANN on the 2nd generation MIT Cheetah footpad, the convergence speed for the estimator of the normal and shear forces is improved by 59.2% compared to using only the neural network alone. The normalized root mean squared error between the two methods are nearly identical at 1.17% in the normal direction, and 8.30% and 10.14% in the shear directions. This approach could have broader implications in greatly reducing the amount of time needed to train a contact force estimator for a large number of tactile sensor arrays (i.e. in robotic hands and skin).**

I. INTRODUCTION

In the field of tactile robotics, there is currently a trend towards using ever larger number of tactile sensors on a robot [1], and this results in a communications bandwidth issue between the sensors and the high-level controller. Driven by the growth of the mobile phone industry, micro-electro-mechanical system (MEMS) sensors have never been cheaper or more readily available. This has led to researchers incorporating large number of sensors into robotic hands for grasping and manipulation [2]–[6], or as a robotic skin in order to measure contact with the external environment to create human-safe robots [7], [8].

However, with such a large number of sensors, there arises a bandwidth issue as it is not feasible to communicate with each individual sensor simultaneously due to the complexity of wiring directly to each sensor as compared to a serial, daisy-chain approach [9], [10]. Most researchers rely on just serially reading each sensor signal, one-by-one, while other researchers have devised schemes such as ”adaptive sensing” [11]. For a large array of tactile sensors, this can lead to an undesirably large delay in data transmission (∼90ms in the case of [8]). Other researchers working on robotic skin have tried to overcome this problem by having either individual or groups of sensor elements have their own dedicated microcontroller for handling signal processing and communications [9], [12], [13]. Nevertheless these methods only help in getting the sensor data to the high-level processor, and do not offer any solutions in reducing the amount of data sent. One way of addressing the problem of using a large number of sensors is to first correlate the data in a meaningful manner locally on dedicated microcontrollers, and only then sending the dimensionally-reduced but relevant data onward to the high-level controller.

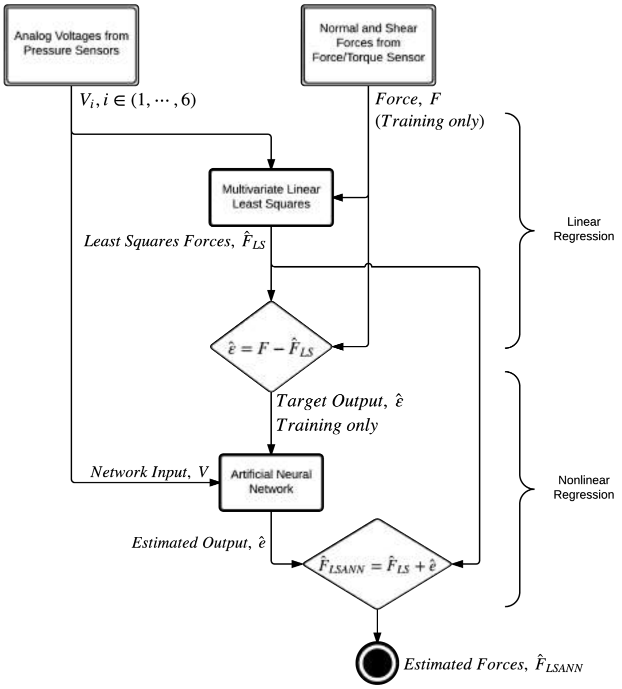

Analog Voltages from Pressure Sensors
Normal and Shear Forces from Force/Torque Sensor
$V_i, i \in (1, \cdots, 6)$
Force, F
(Training only)
Multivariate Linear Least Squares
Least Squares Forces, $\hat{F}_{LS}$
Linear Regression
$\hat{\varepsilon} = F - \hat{F}_{LS}$
Target Output, $\hat{\varepsilon}$
Training only
Network Input, V
Artificial Neural Network
Nonlinear Regression
Estimated Output, $\hat{\varepsilon}$
$\hat{F}_{LSANN} = \hat{F}_{LS} + \hat{\varepsilon}$
Estimated Forces, $\hat{F}_{LSANN}$

Fig. 1. **Flowchart for the Training of the Least Squares Artificial Neural Network (LSANN).** Data collected from both the footpad sensor and the force/torque sensor is used to generate the least squares estimator, which is then used to train and cross-validate the LSANN. Once the LSANN has been trained, it can be use to generate force estimates from the footpad sensor voltages alone.

Correlating the sensor signals both spatially and temporally is useful as it can be used to determine more complex tactile interactions, such as slip [14], [15], twist [16], pinch [17], and shear [18]–[20]. Researchers in the field of underwater robotics have correlated signals from an array of pressure sensors for use as lateral-line sensors for navigation [21]. However with many of these sensors, a Fourier transform is used to correlate the signals, meaning that it can take time to process the sensor signals (∼17ms

[1]Authors are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA. Corresponding email: mcx@mit.edu

This work was supported by the Defense Advanced Research Projects Agency (DARPA) Maximum Mobility and Manipulation (M3) program.

[2]Individual fellowship support was provided by the Agency for Science, Technology and Research (A*STAR) in Singapore.

in the case of slip detection [14]). In the context of robots that run at high speeds such as the MIT Cheetah, where the ground contact time is ∼110ms, a large sensor delay or latency might mean that it will be too late for the robot to take corrective actions. This highlights the importance of managing the communications bandwidth to keep the latency low.

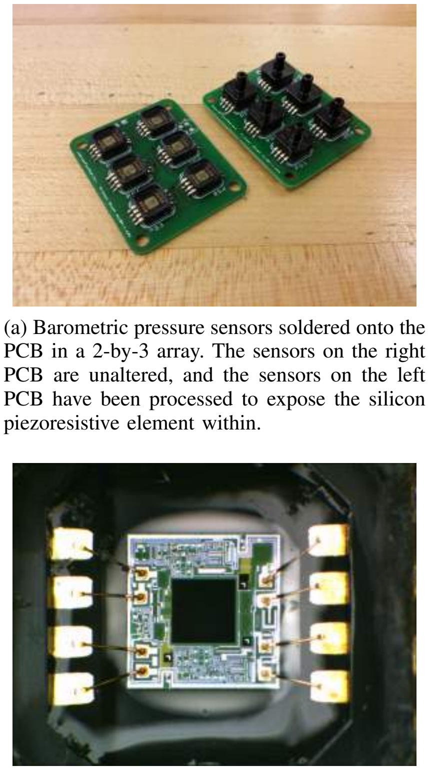
Towards the goal of obtaining real-time (1kHz) normal and shear force estimates for use in robotic ground locomotion, a footpad sensor prototype has been developed for the MIT Cheetah [19], [20]. The footpad takes the readings of an array of piezoresistive sensors and uses that to reconstruct the ground interaction forces. In a future prototype, a dedicated microcontroller will do this processing of the individual sensor readings locally, and only send the relevant three-axis force measurement up to the high-level controller. This would help alleviate the communications bandwidth issue mentioned earlier as it reduces the amount of data being sent. While the sensor performance was satisfactory, the convergence rate during training of the neural network used to obtain the normal and shear force correlation was slow.

To speed up the sensor characterization, it has been split into two parts: a linear regression portion using multivariate least squares regression, and a nonlinear regression portion using a neural network as a multi-input, multi-output function approximator. This procedure has been termed Least Squares Artificial Neural Network (LSANN) and will be covered in more detail in Section III. Fig. 1 shows a flowchart of how the footpad sensor signals and the force/torque sensor data is used to train the LSANN estimator. The force/torque sensor readings are only used to train the LSANN in a one-time process. Once the LSANN has been trained, it can be use to generate force estimates from the footpad sensor signals alone. Other tactile sensors that use neural networks for characterization [3], [22], [23] could possibly benefit from implementing LSANN. In applications such as robotic hands, there could be a large number of tactile sensor arrays (perhaps one in each phalanx or link of the fingers), and so this approach would greatly expedite the convergence of the contact force estimators in the many sensor arrays, reducing the burden on the communications line.

II. Footpad Sensor Architecture

A. *Sensor Electronics*

This footpad is an improvement on the previous design described by Chuah et al. [19], [20] where the goal is to build a lightweight, low cost, yet robust footpad sensor suitable for use in legged robots undergoing ground locomotion. A 2-by-3 array of barometric pressure sensors (MPXH6400A from Freescale Semiconductor) is mounted onto a custom printed circuit board (PCB) of size 32mm by 39mm. This is a reduction in size from the previous prototype of 40mm by 50mm. The thermoplastic casing on the pressure sensors have been machined off to expose the silicon piezoresistive
?? element within. The difference between altered and unaltered sensors can be seen in Fig. 2a and a magnified view of the exposed silicon piezoresistive element can be seen in Fig. 2b.


(a) Barometric pressure sensors soldered onto the PCB in a 2-by-3 array. The sensors on the right PCB are unaltered, and the sensors on the left PCB have been processed to expose the silicon piezoresistive element within.

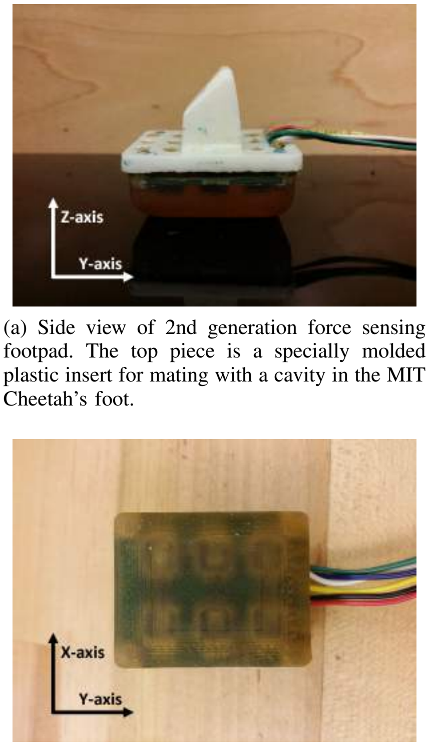
(b) Closeup view of the exposed silicon piezoresistive transducer within a barometric pressure sensor. The black square in the middle is the micromachined silicon diaphragm covered by a silicone gel.

Fig. 2. **Barometric pressure sensor array based force sensor.**

This was done to improve the sensitivity of the sensors when embedded in polyurethane rubber as described in Section II-B]. As the barometric pressure sensors are commercial off-the-shelf (COTS) sensors intended for measuring air pressure, it is necessary to alter them in this manner to achieve the desired sensitivity to deformations within the polyurethane rubber. The analog signals from the sensors are passed through a 12-bit Analog-to-Digital Converter (ADC) (MCP3208 from Microchip Technology) that has been integrated onto the PCB to prevent noise issues. The ADC then interfaces with a 32-bit ARM Cortex-M3 microcontroller (LPC1768 from NXP Semiconductors) through the Serial Peripheral Interface (SPI) bus. The microcontroller then processes the data before it is passed onto the CPU in the MIT Cheetah.

B. *Design and Fabrication*

The design of the footpad has also been improved upon and made more robust. The PCB with the associated electronics described in Section II-A is mounted onto a custom insert made out of rigid, white urethane plastic (Task 4 urethane casting resin from Smooth-On) that is purpose made to fit into a cavity in the MIT Cheetah's foot, hence the unique geometry. This can be seen in Fig. 3a. With the PCB mounted onto the urethane plastic insert, the footpad is then placed into a mold of the desired geometry and the



(a) Side view of 2nd generation force sensing footpad. The top piece is a specially molded plastic insert for mating with a cavity in the MIT Cheetah's foot.

(b) Bottom view of the 2nd generation force sensing footpad. The sensors and PCB are embedded in the translucent polyurethane rubber material.

Fig. 3. **2nd generation of barometric pressure sensor array based force sensor.**

barometric pressure sensors are over-molded with a translucent polyurethene rubber of Shore A hardness 20 (Vytaflex 20 from Smooth-On). This can be seen in Fig. 3b. During the rubber molding process, the sensors are coated with the uncured polyurethane rubber and placed into a vacuum chamber to draw out and remove any air bubbles between the barometric pressure sensors and the polyurethane rubber. The liquid polyurethane rubber is then allowed to cure for 16 hours and solidify. This results in a completely monolithic footpad sensor that is robust, and protected from the external environment, while weighing only 44g.

III. Sensor Characterization via Least Squares Artificial Neural Network (LSANN)

Using the Least Squares Artificial Neural Network (LSANN) procedure, linear portions of the footpad sensor data is first largely fitted using multivariate least squares regression. This gives us a linear estimate of the normal and shear forces that generally shows the trends in the data, but fails to capture nonlinearities where rapid transitions occur. The remaining nonlinearities in the data can then be expressed as the difference between the linear force estimates, and the actual force/torque sensor force readings. This difference is then fitted using a neural network as a multi-input, multi-output function approximator to obtain a nonlinear estimate of the difference. The LSANN estimate of the normal and shear force is then the sum of the linear estimate and the nonlinear differences. Once the LSANN estimator has been trained with a set of force/torque sensor data, it does not need to be re-trained and it is a one-time process. Subsequent normal and shear force estimates are obtained using only the signals from the footpad sensor as inputs.

Such as?

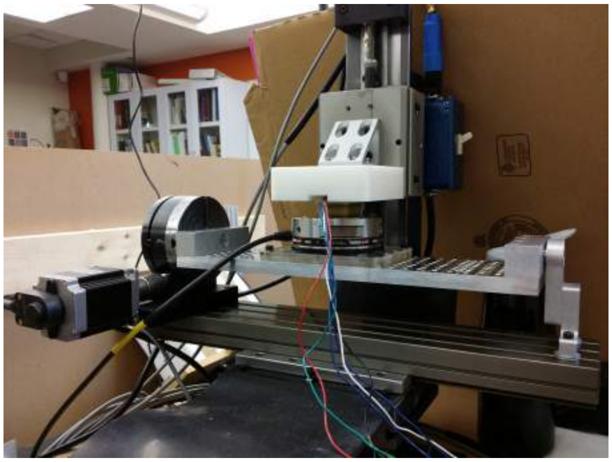


Fig. 4. **Data collection using the CNC mill and a force/torque sensor.** The footpad sensor is mounted to the CNC spindle mount and brought into contact with the force/torque sensor. A prescribed trajectory is then assigned, and data is collected using NI LabVIEW.

A. *Experimental Setup for Data Collection*

In order to make sense of the individual analog voltages collected from the sensors, this data has to be correlated with actual force readings. In this case, a 6-axis force/torque sensor (ATI Delta SI-660-60 from ATI Industrial Automation) is used as the ground truth. A 3-axis CNC milling machine (MicroMill DSLS 3000 from MicroProto Systems) was altered and programmed to act as a positioner and to bring the footpad sensor into contact with the force/torque sensor and move it through different trajectories. This is the same procedure used previously in [19], [20]. Fig. III-A shows the experimental setup with the CNC mill pressing the footpad sensor onto the force/torque sensor. During the motion of the trajectories, 12-bit analog voltages from the six barometric pressure sensors embedded in the footpad are read via SPI by a ARM Cortex-M3 microcontroller a 1kHz sampling rate. At the same time, the forces in the normal and shear directions of the force/torque sensor are collected through a data acquisition system (CompactDAQ 9205 from National Instruments (NI)) at 1kHz as well, and both the voltages from the footpad and the forces from the force/torque sensor are synchronized in LabVIEW. During testing before the data collection, no discernible lag (<1ms) was ever found. The data is then further processed in MATLAB using the LSANN procedure described below.

B. *Linear Estimator using Least Squares Regression*

Multivariate linear least squares regression is used to pre-process the raw voltage signals and produce a linear estimator of the forces in each of the normal and shear axis. This step has the added benefit of reducing the dimension of the input from six analog voltage signals down to three estimated force signals. This is done by first collecting data in each of the different axis of interest using the CNC mill. The analog voltages are assembled into the array below where $v_{n,i}$ represents the n^{th} voltage sample from the i^{th} pressure sensor, where $i \in (1,..,6)$ and $n \in (1,..,N)$. v_i is a column vector of N samples.

INPUT DATA

$$V = \begin{pmatrix} \vdots & & \vdots & & \vdots \\ v_1 & \cdots & v_i & \cdots & v_6 \\ \vdots & & \vdots & & \vdots \end{pmatrix} = \begin{pmatrix} v_{1,1} & \cdots & v_{1,i} & \cdots & v_{1,6} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ v_{n,1} & \cdots & v_{n,i} & \cdots & v_{n,6} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ v_{N,1} & \cdots & v_{N,i} & \cdots & v_{N,6} \end{pmatrix} \quad (1)$$

The corresponding force vectors are given as column vectors below where $f_{g,n}$ is the n^{th} force measurement in the g^{th} axis, where $g \in (X,Y,Z)$ and $n \in (1,..,N)$. $F_g = (f_{g,1} \cdots f_{g,n} \cdots f_{g,N})^T$ is a column vector of N samples.

Using the multivariate linear regression model, we assume that the force in each axis is a linear combination of the voltages from the individual sensors multiplied by the parameters, $a_g = (a_{g,1} \cdots a_{g,7})^T$, and this can be represented in the equation below. ε is the residual.

$$F_g = Va_g + \varepsilon \quad (2)$$

We can obtain an estimate of the parameters, $\hat{a}_g$ by minimizing the sum of squares.

$$\hat{a}_g = (V^T V)^{-1} V^T F_g \quad (3)$$

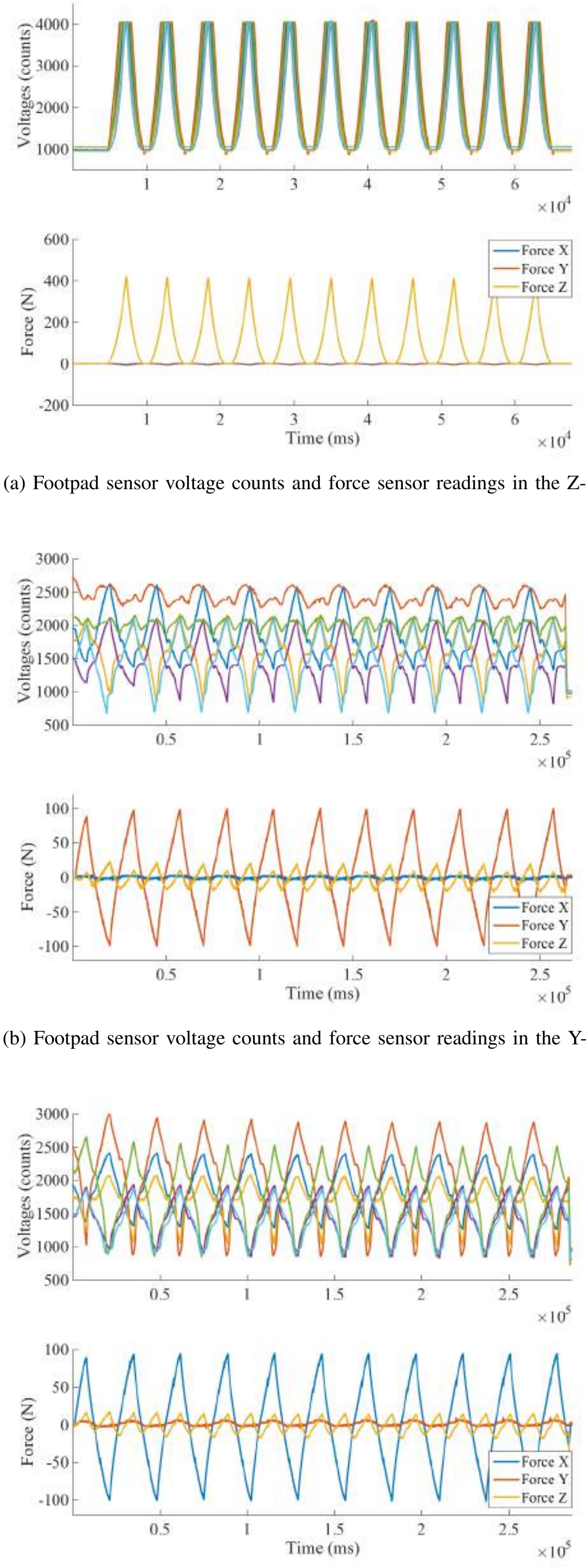
This then allows us to obtain an estimator for the force in each axis using the voltage array in Eqn. 1.

$$\hat{F}_g = V\hat{a}_g \quad (4)$$

Using the CNC mill, the footpad sensor was brought into contact with the force/torque sensor and moved vertically downward in the Z-axis until one of the pressure sensor readings reached saturation. This was then repeated ten times to get the data for computing the least squares parameter for the normal force estimator in the Z-axis, $\hat{a}_Z$. Fig. 5 shows the data collected. More complicated loadings could have been used, but the idea here is to keep the data input largely linear, so that the multivariate least squares regression does not try to latch on to any nonlinearities present in the data.

The same procedure was repeated for the horizontal X- and Y-axis to get the data needed to obtain the least squares parameters for the shear force estimator in the X- and Y-axis, $\hat{a}_X$ and $\hat{a}_Y$.

The least squares force estimator in the Z-axis, $\hat{F}_Z$ gives a root mean squared error (RMSE) of 0.91N out of a range of 221.05N when compared to the true force value, F_Z.



(a) Footpad sensor voltage counts and force sensor readings in the Z-axis

(b) Footpad sensor voltage counts and force sensor readings in the Y-axis

(c) Footpad sensor voltage counts and force sensor readings in the X-axis

Fig. 5. **Footpad sensor voltage counts and force sensor readings.** Data collected for the multivariate linear least squares regression. The footpad sensor is moved forwards and backwards in each axis repeatedly.

This gives a normalized RMSE of 0.42%. For the Y-axis, $\hat{F}_Y$ gives a RMSE of 13.34N out of a range of 233.35N for a normalized RMSE of 5.72% and for the X-axis, $\hat{F}_X$ gives a RMSE of 21.09N out of a range of 218.99N for a normalized RMSE of 9.63%. Overall, this indicates an excellent fit in each axis to the linear training data. As a point of comparison, soft silicon pressure sensors developed by De Rossi et al. for use in a physical human-robot interface of a lower-limb exoskeleton estimates normal forces with normalized RMSE that range from 2.7% to 8.5% [24].

$$RMSE = \left[\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2\right]^{\frac{1}{2}} \tag{5}$$

$$Normalized\ RMSE = \frac{RMSE}{Range\ of\ F_g} * 100\% \tag{6}$$

C. Nonlinear Regression using a Neural Network

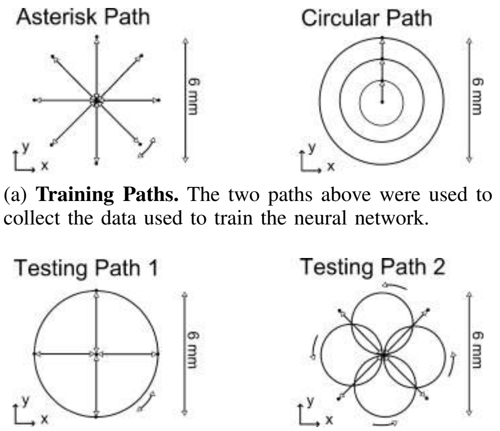
Having obtained a good force estimator through multivariate linear regression, we now seek to improve the estimator by incorporating the nonlinear portions of the data as well. We take the difference between the actual force data, F and our estimated force data, $\hat{F}_{LS}$, as the desired output to our neural network, where the input is the array of voltage signals from the barometric pressure sensors. This makes use of the universal approximation property of neural networks to pose a multi-input-multi-output nonlinear modeling task.

First, we need to collect more data that is used to train the neural network. Training data was first collected by having the CNC milling machine run through a programmed training path with the footpad in contact with the force/torque sensor. Under a range of displacements in Z-axis up to 1.5mm with steps of 0.1mm, the footpad was made to follow the 'Asterisk Path' and 'Circular Path' as shown in Fig. 6a. The 'Asterisk Path' starts from the origin and traverses outwards to 3mm and then back to the origin. This repeats at 45 degree intervals until a full revolution is made. The 'Circular Path' involves the footpad traversing a 2mm, then 4mm, and finally a 6mm diameter circle in both clockwise and counter-clockwise directions.

For cross-validation, a separate set of data has to be collected to test the trained neural network on. 'Testing Path 1' involves the footpad moving 3mm along both the positive and negative X-axis, followed by the Y-axis and ending with it traversing a circular path of 6mm diameter clockwise and then counter-clockwise. 'Testing Path 2' involved a diagonal motion of 3mm in each of the 4 quadrants of the X- and Y-axis. This was then followed with 4 smaller circular paths of 3mm diameter along each of the positive and negative X- and Y-axis. A qualitative depiction of the testing paths is shown in Fig. 6b.

The Levenberg-Marquardt (LM) algorithm [25]–[27] is then used to train a feedforward neural network with a hidden layer size of 10, and it is given as:

$$[J^T W J + \lambda \operatorname{diag}(J^T W J)]\delta = J^T W[\hat{\varepsilon} - \hat{e}] \tag{7}$$



(a) **Training Paths.** The two paths above were used to collect the data used to train the neural network.

(b) **Testing Paths.** The two paths above were used to collect the data used to test the neural network.

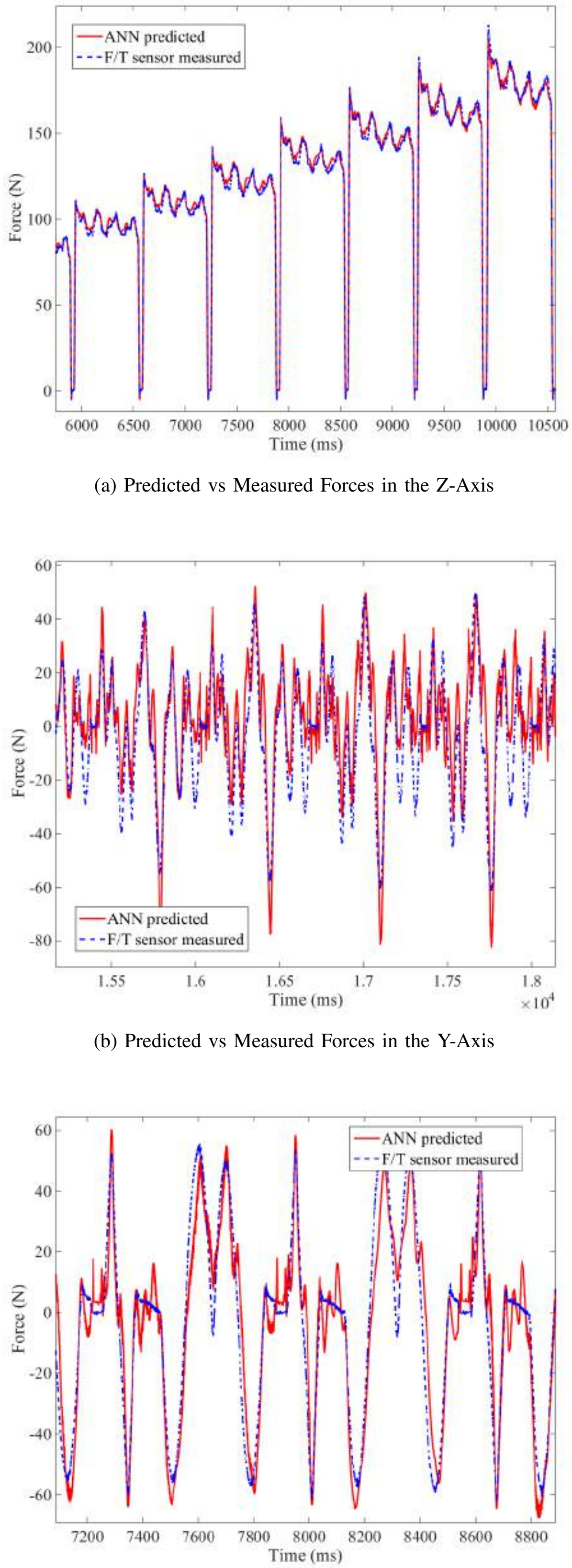
Fig. 6. Training and testing paths used to collect the data needed for cross-validation. Both paths were run at several, set normal displacements in the Z-axis.

where J is the Jacobian matrix, W is the weighting matrix, λ is the algorithmic parameter, δ is the increment in each iteration. The target force output is set as the residual, $\hat{\varepsilon}$, or difference between the actual force data, F and our estimated force data, $\hat{F}_{LS}$, i.e. $\hat{\varepsilon} = F - \hat{F}_{LS}$. When fully trained, the neural network gives us an estimate of the nonlinear differences between the actual forces and the linear force estimates, $\hat{e}$. To get the combined force estimate, $\hat{F}_{LSANN}$, we simply sum the linear force estimate, $\hat{F}_{LS}$, and the nonlinear difference estimate, $\hat{e}$, to get the equation below.

$$\hat{F}_{LSANN} = \hat{F}_{LS} + \hat{e} \tag{8}$$

Using the cross-validation data collected from the 'Testing Path 1' and 'Testing Path 2' trajectories. We can see how well our LSANN estimator is at approximating the normal and shear forces. By calculating the RMSE, normalized RMSE, and coefficient of determination, R^2, we see that LSANN gives a near perfect estimate of the Z-axis force, with a normalized RMSE of only 1.17% and a R^2 value of 99.78%. For the shear directions, the X- and Y-axis, the fit is less perfect, with a normalized RMSE of 8.30% and a R^2 value of 81.03% for the X-axis, and a normalized RMSE of 10.14% and a R^2 value of 72.22% in the Y-axis. This compares favorably with the estimator obtained using just the neural network as a function fitting approximator. As an additional benefit, the time needed to train the LSANN is only 59.0 minutes, while for just the neural network, the training time is 144.5 minutes. This indicated a training time reduction of 59.2%. The estimator is a better fit in the X-axis compared to the Y-axis due to the barometric pressure sensor array being in a 2-by-3 configuration. This data is represented in Table I and Fig. 7 shows the agreement between the LSANN predicted forces and the actual force/torque sensor readings.

$$R^2 = \left[1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}\right] * 100\% \tag{9}$$



(a) Predicted vs Measured Forces in the Z-Axis

(b) Predicted vs Measured Forces in the Y-Axis

(c) Predicted vs Measured Forces in the X-Axis

Fig. 7. **Experimental results for agreement between the predicted force and actual force.** The blue line shows the actual force measured and the red line shows the LSANN predicted force. Only a portion of the data collected is shown here for the sake of clarity.

TABLE I

RMSE, NORMALIZED RMSE, AND THE COEFFICIENT OF DETERMINATION, R^2 FOR EACH AXIS

Z-axis	RMSE (N)	Normalized RMSE (%)	$R^2(\%)$
LSANN	2.58	1.17	99.78
Neural Network only	2.42	1.10	99.90

Y-axis	RMSE (N)	Normalized RMSE (%)	$R^2(\%)$
LSANN	12.84	10.14	72.22
Neural Network only	12.63	9.98	79.02

X-axis	RMSE (N)	Normalized RMSE (%)	$R^2(\%)$
LSANN	10.48	8.30	81.03
Neural Network only	11.17	8.85	83.62

IV. CONCLUSIONS

This paper presents Least Squares Artificial Neural Network (LSANN) as an approach to the characterization of tactile array sensors for the reduction of the computational time needed for convergence to obtain a useful estimator for normal and shear forces. Sensor characterization is split into a linear regression portion, and a nonlinear regression portion. The linear regression portion applies a multivariate least squares regression to data captured in each of the normal and shear directions. This gives us a linear estimator for the forces, $\hat{F}_{LS}$. The nonlinear regression portion uses a neural network with the Levenberg-Marquardt algorithm as a multi-input, multi-output function approximator, with the target output being the residual, $\hat{\varepsilon}$, or difference between the actual force data, F and our estimated force data, $\hat{F}_{LS}$, i.e. $\hat{\varepsilon} = F - \hat{F}_{LS}$. Once fully trained, the neural network gives us an estimate of the nonlinear differences, $\hat{e}$, between the actual forces and the linear force estimates. To get the combined force estimate, $\hat{F}_{LSANN}$, we simply sum the linear force estimate, $\hat{F}_{LS}$, and the nonlinear difference estimate, $\hat{e}$, to get $\hat{F}_{LSANN} = \hat{F}_{LS} + \hat{e}$.

By applying LSANN on the 2nd generation MIT Cheetah footpad, the convergence speed for the estimator of the normal and shear forces is improved by 59.2% compared to using only the neural network alone from 144.5 minutes down to 59.0 minutes. While the convergence time of 59.0 minutes is relatively long, this only needs to be done once and the resulting parameters can be stored and reused. The convergence time could also be shortened even more by using a computer with a faster processor to do the calculations. The normalized root mean squared error between the two methods are nearly identical at 1.17% in the normal direction, and 8.30% and 10.14% in the shear directions. There is ongoing work to improve the shear direction readings by using the results of finite element analysis simulations.

This approach could have broader implications in greatly reducing the amount of time needed to train an estimator for tactile sensor arrays where the number of sensor elements are large, as a large number of inputs to a neural network generally increases the computational time needed for convergence. For example in robotic hands or skin, where multiple sensor arrays for determining normal and shear contact forces during tactile interactions are desired, having *n* sensor arrays would mean that using the LSANN approach would make the characterization process $\sim 0.6n$ times faster. This can also speed up the process of determining more complex tactile interactions by making it faster to correlate the sensor signals both spatially and temporally, potentially making it easier for robots to detect slip and shear events as well as other tactile modalities. There is currently ongoing work to implement the LSANN computation on a dedicated microcontroller to evaluate the computational time in real-time operation.

ACKNOWLEDGMENT

Meng Yee (Michael) Chuah would like to thank Albert Wang and Patrick Wensing of the MIT Biomimetic Robotics Laboratory for the many fruitful discussions that helped shape this work.

REFERENCES

[1] M. Lee and H. Nicholls, "Tactile sensing for mechatronics: a state of the art survey," pp. 1–31, 1999.

[2] N. A. Radford, P. Strawser, K. Hambuchen, J. S. Mehling, W. K. Verdeyen, A. S. Donnan, J. Holley, J. Sanchez, V. Nguyen, L. Bridgwater, R. Berka, R. Ambrose, M. Myles Markee, N. J. Fraser-Chanpong, C. McQuin, J. D. Yamokoski, S. Hart, R. Guo, A. Parsons, B. Wightman, P. Dinh, B. Ames, C. Blakely, C. Edmondson, B. Sommers, R. Rea, C. Tobler, H. Bibby, B. Howard, L. Niu, A. Lee, M. Conover, L. Truong, R. Reed, D. Chesney, R. Platt, G. Johnson, C.-L. Fok, N. Paine, L. Sentis, E. Cousineau, R. Sinnet, J. Lack, M. Powell, B. Morris, A. Ames, and J. Akinyode, "Valkyrie: NASA's First Bipedal Humanoid Robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, May 2015.

[3] R. Balasubramanian and V. J. Santos, THE HUMAN HAND AS AN INSPIRATION FOR ROBOT HAND DEVELOPMENT. Springer International Publishing, 2014, vol. 95.

[4] L. P. Jentoft, Q. Wan, and R. D. Howe, "Limits to compliance and the role of tactile sensing in grasping," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2014, pp. 6394–6399.

[5] H. Kawasaki, T. Komatsu, and K. Uchiyama, "Dexterous anthropomorphic robot hand with distributed tactile sensor: Gifu hand II," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 3, pp. 296–303, Sep. 2002.

[6] P. Dario, A. Sabatini, B. Allotta, M. Bergamasco, and G. Buttazzo, "A fingertip sensor with proximity, tactile and force sensing capabilities," *IEEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*, pp. 883–889, 1990.

[7] S. Phan, Z. F. Quek, P. Shah, D. Shin, Z. Ahmed, O. Khatib, and M. Cutkosky, "Capacitive skin sensors for robot impact monitoring," *IEEE International Conference on Intelligent Robots and Systems*, no. Figure 1, pp. 2992–2997, 2011.

[8] T. Mukai, S. Hirano, and Y. Kato, "Fast and Accurate Tactile Sensor System for a Human-Interactive Robot," *cdn.intechweb.org*, no. December, 2008.

[9] L. Ascari, P. Corradi, L. Beccai, and C. Laschi, "A miniaturized and flexible optoelectronic sensing system for tactile skin," *Journal of Micromechanics and Microengineering*, vol. 17, no. 11, pp. 2288–2298, Nov. 2007.

[10] J. Tegin and J. Wikander, "Tactile sensing in intelligent robotic manipulation: a review," pp. 64–70, 2005.

[11] J. Jockusch, J. Walter, and H. Ritter, "A tactile sensor system for a three-fingered robot manipulator," *Proceedings of International Conference on Robotics and Automation*, vol. 4, 1997.

[12] P. Mittendorfer and G. Cheng, "Humanoid multimodal tactile-sensing modules," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 401–410, 2011.

[13] F. Cuellar, T. Yamamoto, and H. Ishiguro, "Design and development of a low power Tactile Multi-Sensor Network for robotic systems," in *2014 IEEE International Conference on Mechatronics and Automation*. IEEE, Aug. 2014, pp. 331–336.

[14] D. Goeger, N. Ecker, and H. Woern, "Tactile sensor and algorithm to detect slip in robot grasping processes," *2008 IEEE International Conference on Robotics and Biomimetics, ROBIO 2008*, pp. 1480–1485, 2008.

[15] B. Heyneman and M. R. Cutkosky, "Slip interface classification through tactile signal coherence," in *IEEE International Conference on Intelligent Robots and Systems*, no. SynTouch LLC, 2013, pp. 801–808.

[16] W. Yuan, "Tactile measurement with a GelSight sensor," Master's thesis, Massachusetts Institute of Technology, 2014.

[17] K. Sato, K. Kamiyama, N. Kawakami, and S. Tachi, "Finger-shaped GelForce: Sensor for measuring surface traction fields for robotic hand," *IEEE Transactions on Haptics*, vol. 3, no. 1, pp. 37–47, 2010.

[18] A. Cirillo, P. Cirillo, G. De Maria, C. Natale, and S. Pirozzi, "An artificial skin based on optoelectronic technology," *Sensors and Actuators A: Physical*, vol. 212, pp. 110–122, Jun. 2014.

[19] M. Y. Chuah, M. Estrada, and S. Kim, "Composite force sensing foot utilizing volumetric displacement of a hyperelastic polymer," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2012, pp. 1963–1969.

[20] M. Y. Chuah and S. Kim, "Enabling Force Sensing During Ground Locomotion: A Bio-Inspired, Multi-Axis, Composite Force Sensor Using Discrete Pressure Mapping," *IEEE Sensors Journal*, vol. 14, no. 5, pp. 1693–1703, May 2014.

[21] A. G. P. Kottapalli, M. Asadnia, J. Miao, and M. Triantafyllou, "Soft polymer membrane micro-sensor arrays inspired by the mechanosensory lateral line on the blind cavefish," *Journal of Intelligent Material Systems and Structures*, vol. 26, no. 1, pp. 38–46, 2014.

[22] G. Palli, C. Melchiorri, G. Vassura, G. Berselli, S. Pirozzi, C. Natale, G. De Maria, and C. May, *Innovative technologies for the next generation of robotic hands*. Springer Berlin Heidelberg, 2012, vol. 80, no. STAR.

[23] A. Ananthanarayanan, S. Foong, and S. Kim, "A Compact Two DOF Magneto-elastomeric Force Sensor for a Running Quadruped," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*. Ieee, May 2012, pp. 1398–1403.

[24] S. M. M. De Rossi, N. Vitiello, T. Lenzi, R. Ronsse, B. Koopman, A. Persichetti, F. Vecchi, A. J. Ijspeert, H. van der Kooij, and M. C. Carrozza, "Sensing pressure distribution on a lower-limb exoskeleton physical human-machine interface." *Sensors (Basel, Switzerland)*, vol. 11, no. 1, pp. 207–27, Jan. 2011.

[25] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[26] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, pp. 164 – 168, 1944.

[27] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm." *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–93, Jan. 1994.