2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)
Beijing, China, November 6-9, 2018

Robust Affordable 3D Haptic Sensation via Learning Deformation Patterns

Huanbo Sun[1] and Georg Martius[1]

***Abstract*— Haptic sensation is an important modality for interacting with the real world. This paper proposes a general framework of inferring haptic forces on the surface of a 3D structure from internal deformations using a small number of physical sensors instead of employing dense sensor arrays. Using machine learning techniques, we optimize the sensor number and their placement and are able to obtain high-precision force inference for a robotic limb using as few as 9 sensors. For the optimal and sparse placement of the measurement units (strain gauges), we employ data-driven methods based on data obtained by finite element simulation. We compare data-driven approaches with model-based methods relying on geometric distance and information criteria such as Entropy and Mutual Information. We validate our approach on a modified limb of the "Poppy" robot [1] and obtain 8 mm localization precision.**

I. Introduction

We are witnessing a rapid development of robot technologies. Actuators and sensors have become increasingly compact and powerful. Nevertheless, robots are still far from matching human capabilities especially when it comes to touch sensation. Haptic information is, however, essential for a reliable interaction with the real world. It becomes evident that robots need to learn interaction patterns for mastering the real world challenges. For this, haptic sensors have to be robust in order to sustain long-lasting experiments. Besides robustness, another important aspect of robotic hardware is its price, availability, and performance. A low cost makes robotic technologies widely accessible and thus facilitates research.

Currently available **large area** haptic sensor systems [2], [3] are expensive, complicated to integrate, and not robust enough to sustain long-term use. Array shaped sensors [4] can localize stimulations but have large amounts of elements and require many wires. To reduce the hardware complexity, methods such as anisotropic electrical impedance tomography (EIT) [5] have been proposed. However, both types are generally not robust because they cover the surface as a skin which makes them vulnerable to impacts with hard or sharp objects. A tiny crack can destroy the functionality of the entire sensor. **Small scaled** sensors, such as BioTac® [6], [7] are already widely used in robotic hand applications. However, their functional area is only at the tip of the finger instead of the whole 3D surface. TacTip [8] is an optical tactile sensor for hollow robotic parts of different sizes. It is able to detect contacting object shapes very accurately while force's information cannot be directly obtained.

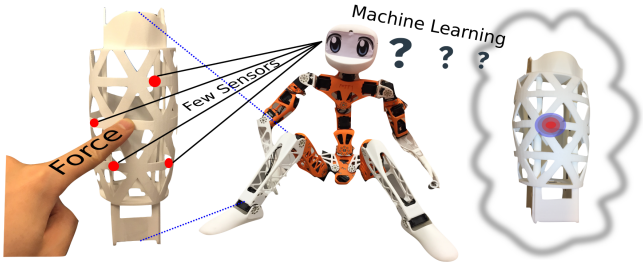

Fig. 1: Overall goal of the method: Inferring single haptic stimulation from few sensors measuring internal deformations.

In this paper, we aim at providing a low-cost, robust and sufficiently precise method for inferring haptic forces on the surface of a 3D structure. Instead of relying on a dense array of sensors on the surface of the robot, we opt for a small number of physical sensors measuring internal deformations. This offers a couple of conceptual advantages. First, the system is robust to environmental impacts because the sensors can be placed inside of the structure. Second, the surface shape can be freely designed. Third, only a few channels have to be read out which reduces both the energy consumption and the data rate.

On the downside, a measurement of the sensors does not directly correspond to the impacting force. Instead, an inference mechanism is required to estimate the force. We propose a data-driven approach using machine learning algorithms to perform this inference efficiently. In order to require as few sensors as possible, we employ several optimization schemes to determine optimal sensor placement.

The contributions of the paper are as follows. On the theory side, we:

- propose a new way of implementing a whole surface haptic sensor,
- provide a method for determining the optimal number and position of sensors using finite element simulations.

On the application side, we

- provide an assembly method for attaching the strain gauges,
- designed a hardware system to systematically collect data,
- demonstrated the sensing system on a robotic limb.

The paper is structured as follows: in Section II, we present the method by first giving an overview and then

*This work was supported by Max Planck Institute for Intelligent Systems
[1]Huanbo Sun and Georg Martius are with the Max Planck Institute for Intelligent Systems, Tübingen, 72076, Germany {huanbo.sun | georg.martius}@tuebingen.mpg.de

investigating optimal sensor placement. In Section III, we present the results on the robotic limb. We close with a discussion in Section IV.

II. Method

We propose a method to implement a whole surface haptic force sensor using a minimal amount of deformation sensors (strain gauges) inside of a 3D structure. Using machine learning, the haptic forces are inferred from the few measured deformations, as illustrated in Fig. 1. We assume that the surface is constructed such that it has an inside rigid support and a flexible outer shell.

In order to place the sensors optimally, we need to get access to data describing how forces applied to the structure propagate into deformations measurable by the sensors. We do this by finite element simulation. In our case, the ANSYS [9] simulation tool was used, see Fig. 2.

Given the simulated deformation patterns for many force impacts (dataset), the problem can be stated as follows: Let $X \in \mathbb{R}^{M \times N}$ denote the deformation (displacement) of N points on the inside surface of the shell and $p \in \mathbb{R}^{M \times 3}$ the position of the applied force for M different locations. We are looking for a subset of locations A among all possible points N such that the force locations can be well inferred i. e. :

$$\underset{A \subset N}{\arg\min} \left(\mathbb{E}[\|F(X_{\cdot,A}) - p\|_2^2] < \delta \right), \tag{1}$$

where $F(\cdot)$ is a learned mapping function and δ is the tolerated error.

Our approach to approximate $Eq.$ (1) is composed of the following steps:

- collect a dataset of deformations from the finite element simulation, see Section II-A,
- filter the possible sensor locations according to physical constraints, see Section II-B,
- learn a non-linear regression model (SVR) to infer the haptic force positions for unseen stimulation locations, see Section II-C,
- select the number and position of sensors needed for a certain pre-determined accuracy, where we evaluate different techniques in Section II-D,
- validate the prediction quality obtained from differently selected sensor positions, see Section III-A.

Afterwards, we apply the optimal selected sensor positions and inference model on a real robotic limb (Section III).

A. Finite Element Simulation

In order to get the deformation patterns for a certain impact force, we use a finite element simulation of the 3D structure. The ANSYS simulation tool [9] allows importing the 3D-CAD description of the structure to equip with force sensation. In simulation, we can apply forces to every location on the surface and record the deformations of all positions on the structure, discretized in a fine manner. The deformation pattern is illustrated in Fig. 2. For this example structure, we obtain around 4000 different force positions and 3000 different sensor positions.

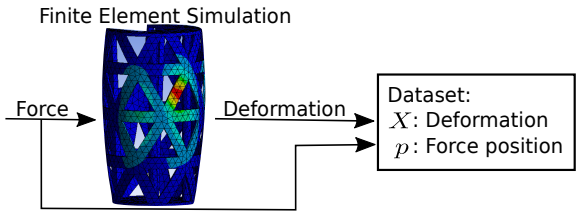

Fig. 2: Data collection using finite element simulation and virtually applied forces.

B. Filtering feasible Sensor Positions

From all positions on the inside of the 3D structure, we need to filter those that allow for physical sensor placement. There are several constraints imposed by the sensor size, their placement restrictions and their range of detection. The strain gauge sensors cannot be placed on edges since they need a relatively flat surface. Placing them near highly rigid support structures is also disadvantageous because only small deformations occur.

In Fig. 3a the unfolded surface of our example structure is displayed. The rigid support is at the top and the bottom of the structure, so we discard positions close to that. In order to get rid of candidate positions at the edges, we use a k Nearest Neighbor (kNN) criterion. For each candidate position, we consider the center of mass of the neighborhood, see Fig. 3b. If the center of mass is inside a certain radius, the position is kept (red points in Fig. 3a). In our examples, we get around 2100 remaining points.

1) Reducing Number of Candidates using Compressive Sensing: To make the optimal selection of sensor positions more efficient, we further reduce the number of candidates. Compressive Sensing techniques [10] can be employed here, which are optimized to reconstruct sparse or compressible signals accurately from a limited number of measurements. A lossless reduction is not possible in our case, but we can bound the maximal tolerated reconstruction error.

We use PCA with QR-Pivoting [11]. At this point, we only give an intuitive understanding and elaborate the details below in Section II-D.2. The method uses PCA to compute the principal components explaining the variance in decreasing order. The QR-pivoting selects those positions (sensors) that are most important for the top principal components.

In Fig. 3c the linear reconstruction error (unexplained variance) in dependence on the number of selected sensor positions is displayed. In our example, for 1% error, we can select 407 out of 2162. The points are also marked in black in Fig. 3a.

So far, we have automatically selected a set of candidate positions.

C. Force position inference (SVR)

From the set of candidate positions, the task is now to find a smaller subset that is sufficient to make the inference about deformations anywhere on the structure. We restrict ourselves in this paper to a method that infers the position of a single impacting force.

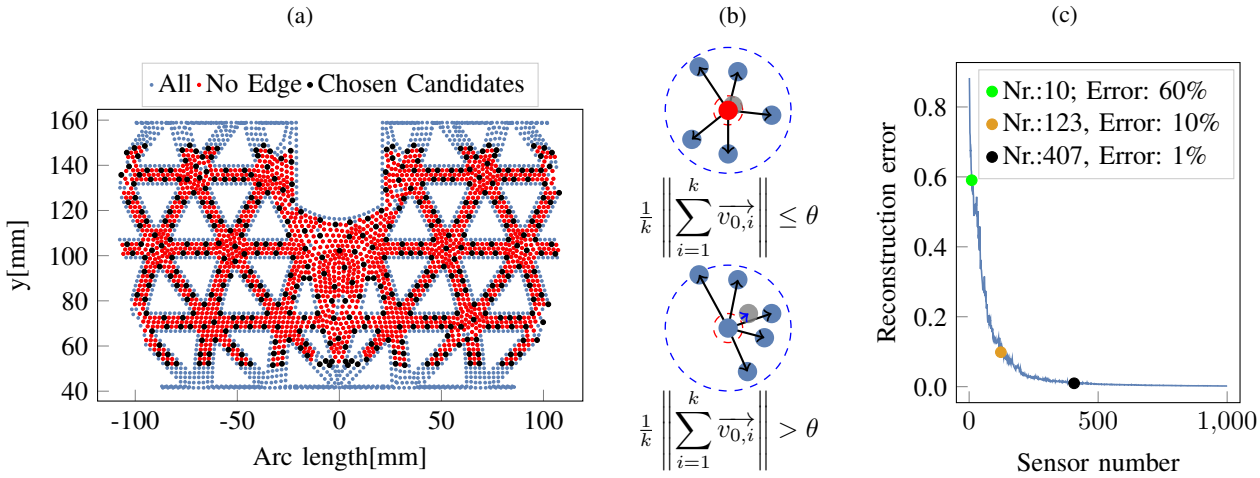

Fig. 3: (a) Unfolded geometry: Sensor positions are indicated with light blue dots and available positions are indicated in red. The most informative chosen positions are indicated in black. (b) KNN method is used to exclude positions that are affected by edge effects. (c) Reconstruct the original data from a small subset of informative sensor positions using compressive sensing method.

Support Vector Machines (SVM) are popular kernel-based algorithms [12] combining the strength of nonparametric techniques with efficient storage requirements of parametrized models. In this paper, we face a regression task such that the Support Vector Regression (SVR) method [13] is applied here. The idea is that the input x is nonlinearly mapped into a high-dimensional feature space where a linear regression with minimum margin is performed.

The model is

$$F(x, w) = \sum_{i=1}^{m} w_i \cdot g_i(x) + w_0 \tag{2}$$

where $g_i(x)$ is the nonlinear feature map, w are the parameters. The regression error for each example x_i is defined as $\xi_i = \max(0, |F(x_i, w) - p_i| - \epsilon)$, i. e. the deviation from the target p_i larger than ϵ. In our case, the input x is the vector of deformations at the selected sensor positions and the target p is the p_x, p_y, p_z position of the applied force point. In fact, we use one SVR model for each of the target dimensions.

SVR is formulated as a minimization of the following function:

$$L = \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i \tag{3}$$

where the hyper-parameter C controls the trade-off between complexity of the regression model (norm of w) and the error.

Interestingly, only scalar products of elements in the feature space are computed, such that one can directly express the scalar product using an appropriate kernel function $k(x_i, x_j) = g(x_i)^\top g(x_j)$. A popular kernel function is the radial basis function (RBF) or Gaussian kernel:

$$k(x_i, x_j) = \exp\left(-\gamma \left\|x_i - x_j\right\|^2\right) \tag{4}$$

with hyperparameter γ controlling the sensitivity to distance. We use the Python *sklearn* [14] implementation. We choose RBF kernels and used k-fold cross-validation to select the optimal C, ε and γ.

As a remark, using all candidate positions and 85% of training data SVR can achieve an average test error of 0.6 mm ($C = 0.1$, $\varepsilon = 10^{-4}$, $\gamma = 2 \times 10^{-3}$, 5-fold cross-validation).

D. Optimal Sensor Placement

In this section, we propose different ways to select sensor positions, which can be generally grouped into data-driven methods and geometry/model-based methods.

Data-driven methods make use of previously collected data and select the subset of sensors with which the regression model performs the best. Geometry/model-based methods do not need access to the measured data. Instead, they rely only on the geometric position of the sensors. If sufficient data is available then the data-driven methods can be more accurate because they have access to the actual dependency between the sensor locations.

In this paper, we propose to use a greedy SVR approach. As a comparison, we provide also results using a linear compressive sensing method and two geometry-based methods.

1) Nonlinear method: Greedy Support Vector Regression: In principle, we want to select this combination of K sensors that perform best on average at inferring the force position for unseen stimulations. The problem is that we would need to search through $\binom{n}{K}$ different possibilities, which is intractable for $K > 4$. Thus, we employ a greedy strategy: Start with the best single sensor position and then add the second sensor position that gives the best performance and so forth. Performance is defined in terms of k-fold cross-validation, see Alg. 1.

2) Linear method: PCA with QR Pivoting: As mentioned in Section II-B.1, the number of sensors can be reduced using methods from compressive sensing. In this section, we provide details about the specific method, namely PCA with QR-Pivoting.

Algorithm 1 Greedy SVR

Input data: Deformation: $X \in \mathbb{R}^{M\times N}$, force position : $y \in \mathbb{R}^{M\times 3}$, maximum sensor budget K
Data standardization & k-fold cross-validation dataset preparation $\{X^1, ..., X^k\}$,$\{y^1, ..., y^k\}$
Selected nodes: $A = \emptyset$
for $i : 1$ **to** K **do**
  for $m : 1$ **to** $M \notin A$ **do**
    for $j : 1$ **to** k **do**
      $A' \leftarrow A \cup m$
      $e_j \leftarrow \|\text{SVR}(X^j\{A'\}) - y^j\|$
    $error_m \leftarrow \text{mean}(e)$
  $A \leftarrow A \cup \arg\min(error)$

Compressive sensing depends on two major functional matrices: feature transform basis $\Psi \in \mathbb{R}^{N\times N}$ and sub-sampling matrix $\Phi \in \mathbb{R}^{m\times N}$. Ψ is designed to transform raw measurements $x \in \mathbb{R}^{N\times 1}$ into a sparse representative space $\alpha \in \mathbb{R}^{N\times 1}$ where α has only s nonzero elements:

$$x = \Psi \cdot \alpha \tag{5}$$

Φ subsamples m measurements from N optimally such that the representation α may be most independently and accurately reconstructed from the measurements $\hat{x} \in \mathbb{R}^{m\times 1}$ using l_1 norm:

$$\alpha = \arg\min_{\alpha'} \|\alpha'\|_1 \qquad \text{s.t.} \quad \hat{x} = \Phi \cdot \Psi \cdot \alpha, \tag{6}$$

where the number of measurements and the sparsity of transformed signal must fulfill the Restricted Isometry Property [15]: $m \gtrsim s{\cdot}log(N)$. Eq. (6) is a linear optimization problem and conditioned on the operator $(\Phi{\cdot}\Psi)$. The central challenge is to design an optimal Ψ compressing raw data x efficiently and find a good Φ such that the operator $(\Phi \cdot \Psi)$ is well-conditioned.

PCA is a linear unsupervised dimension reduction method [16]. It finds the directions of maximum variance in high-dimensional data and projects data onto a smaller dimensional subspace while remaining most of the information. We keep the first s principal components Ψ_s of Ψ to ensure the s sparsity in α and then select an optimal sub-sampling Φ to constrain the reconstruction error of Eq. (6) based on the condition number criterion. The condition number of the operator $(\Phi \cdot \Psi_s)$ is denoted as:

$$c = \frac{\sigma_{max}(\Phi \cdot \Psi_s)}{\sigma_{min}(\Phi \cdot \Psi_s)} \tag{7}$$

Since Φ is a permutation measurement matrix, it can be designed as the column pivoting matrix of Ψ_s. QR factorization with column pivoting is used to select the highest singular values such that σ_{min} is maximized while c is minimized. Details are shown in Alg. 2.

3) Model-based methods using Gaussian Process: We want to compare the data-driven methods with those relying only on the geometric location of the sensors. They assume that the geometry is homogeneous and all sensors have a fixed sensing radius, hence they are called model-based. A convenient model to predict unmeasured sensor values is to use a Gaussian Process (GP) [17], [18]. It models a distribution over functions with a continuous domain, here functions from sensor location to deformations. It uses a similarity between locations (sensors) which is measured by a localized kernel to construct a covariance structure. This in turn is used to make sure that predicted values of similar locations are similar. However, for a new location, GPs predict not only a mean estimate but also the uncertainty represented by a one-dimensional Gaussian distribution.

Algorithm 2 PCA with QR Pivoting

Input data: Deformation Pattern: $X^{M\times N}$, Maximum Sensor Budget K
Data standardization
Principal Components Decomposition $\Psi^{N\times N}$
for $i : 1$ **to** K **do**:
  Pick $1 : i^{th}$ principal components $\Psi^{N\times i}$
  $P \leftarrow$ **QR-Pivoting**$(\Psi^{N\times i})$
  Sensor node marker = $P[0 : i]$

a) Gaussian Process and model/kernel selection: We start with formalizing the prediction procedure of a GP and then choose the right kernel.

Given a set of sensors A, their positions p_A, and their deformation data $X_{A,\cdot}$ we can predict the distribution of deformations at a different sensor y with location p_y. The mean $\mu_{y|A}$ and standard deviation $\sigma_{y|A}$ are given by

$$\mu_{y|A} = \mu_y + \Sigma_{y|A} \cdot \Sigma_{AA}^{-1} \cdot (X_{A,\cdot} - \mu_A) \tag{8}$$

$$\sigma^2_{y|A} = k(y, y) - \Sigma_{y|A} \cdot \Sigma_{AA}^{-1} \cdot \Sigma_{y|A}^T\,, \tag{9}$$

where $\mu_A = \frac{1}{|A|}\sum_{a\in A} X_{\cdot,A}$ is the vector of mean sensor values for the sensors in A, $\Sigma_{AA}(i, j) = k(p_i, p_j) + \beta^{-1}\delta_{i,j}$ is covariance matrix/kernel matrix for all sensors $i, j \in A$ with δ_{ij} being the Kronecker delta and β is a hyperparameter. Similarly, $\Sigma_{y|A}(i) = k(p_i, p_y)$ is the vector of similarities with the new sensor location.

The GP is a non-parametric process after picking the kernel $k(\cdot, \cdot)$. Typical choices are polynomial, rational quadratic, and exponential kernels. In our application, the deformations vary smoothly and locally w. r. t. the location, as they can be described by the bending of a thin plate [19]. Thus, we use an exponential kernel

$$k(x, x') = \exp\left(-\left(\frac{d(x, x')}{l_{scale}}\right)^{l_p}\right) \tag{10}$$

with hyperparameters for the length scale l_{scale} and for the distance norm l_p. The distance is measured by $d(x, x')$ which is the approximate geodesic distance instead of the Euclidean distance because the surface of the 3D structure is curved.

For hyperparameter selection, we use cross-validation [20] which was shown to be robust. The grid search for the parameters l_{scale} and l_p is shown in Fig. 4a. We choose $l_{scale} = 0.033$ and $l_p = 1.9$.

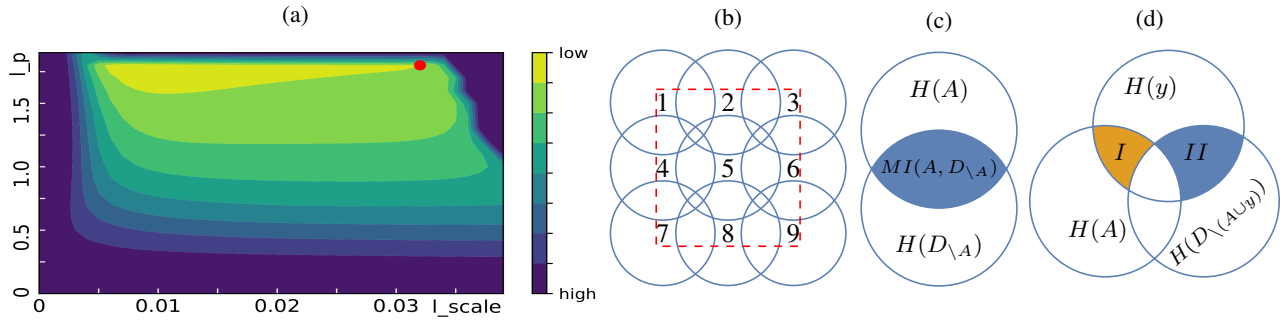

Fig. 4: The average and standard deviation of critical parameters. (a) Model selection for GP covariance. (b) Entropy criterion: Each sensor has the same ability to detect uncertainty indicated as the same radius. Disk 1, 3, 7,9 are first selected as the overlapping area is zero such that the ability to discover total uncertainty is maximized. But those areas outside the red FOV are wasted. (c) The blue shaded area indicates the shared information between selected set A and unselected set $D_{\setminus A}$. (d) Intuitive explanation for Eq. (13): For each step, we choose the sensor y which shares the least information with set A but represents most information of the rest unselected set $D_{\setminus(A\cup y)}$ and generally maximize area $(II - I)$.

The probabilistic modeling of the data allows us to use information criteria for selecting the most informative sensor positions. The first method minimizes the uncertainty about the non-measured locations (Entropy) and the second maximizes the Mutual Information (MI) between selected sensor and the non-measured locations.

As before we select k best sensor positions A out of all permissible locations V (red and black points in Fig. 3a).

4) Pick locations with maximal uncertainty – Entropy: By intuition, a good design is to pick sensor locations that minimize the uncertainty about the entire permissible locations V. This can be quantified by the conditional entropy of the unobserved locations $V_{\setminus A}$ given the observed ones, i. e. $H(V_{\setminus A}|A)$ [21]. Mathematically, we aim at:

$$A^* = \arg\min_{A\subset V} H(V_{\setminus A}|A) = \arg\max_{A\subset V} H(A)\,, \qquad (11)$$

see [21] for details. Since Eq. (11) involves a combinatorial search we solve it greedily, as done in the SVR case in Section II-D.1. The entropy of a Gaussian distribution is analytically given as $H(\mathbb{N}(\mu,\sigma^2)) = \frac{1}{2}\ln(2\pi e\sigma^2)$. The algorithm is detailed in Alg. 3

Algorithm 3 Maximum Uncertainty: Entropy

1: Input data: GP, sensor budget K
2: **for** $i : 1$ **to** K **do**:
3: $\quad y^* \leftarrow \text{argmax}_{y\in V_{\setminus A}} \sigma^2_{y|A}$ using Eq. (9)
4: $\quad A \leftarrow A \cup y^*$

This will automatically choose sensors far away from each other, as illustrated in Fig. 4b. As a side effect, the selected locations tend to sit on the boundary of the space, making them in principle inefficiently using their full detection disk.

5) Mutual Information: Another criterion suggested in [21] is the Mutual Information (MI) which measures the shared information between selected and unselected locations:

$$\text{MI}(A, D_{\setminus A}) = H(D_{\setminus A}) - H(D_{\setminus A}|A), \qquad (12)$$

where D is the set of all locations (also those that are not permissive as sensor location , e. g. all light blue points in Fig. 3a). An intuitive explanation is given in Fig. 4c. Maximizing the MI between $D_{\setminus A}$ and A is also a combinatorial problem, such that a greedy method is used as well. In each step we pick the location with maximum additive Mutual Information:

$$y^* = \arg\max_{y\in V_{\setminus A}} \left[\text{MI}(A\cup y, D_{\setminus A\cup y}) - \text{MI}(A, D_{\setminus A})\right] \qquad (13)$$

as detailed in Alg. 4 and intuitively indicated in Fig. 4d: For each step, we choose the sensor y which shares the least information with set A but represents most information of the rest unselected set $D_{\setminus A\cup y}$ and generally maximize area $(II - I)$.

Algorithm 4 Mutual Information Criterion

1: Input data: GP, sensor Budget K
2: **for** $i : 1$ **to** K **do**:
3: $\quad B(y) = D_{\setminus(A\cup y)}$
4: $\quad y^* \leftarrow \arg\max_{y\in V_{\setminus A}} \left(\frac{\sigma^2_{y|A}}{\sigma^2_{y|B(y)}}\right)$ using Eq. (9)
5: $\quad A \leftarrow A \cup y^*$

III. Results

Using the methods presented above, we first determine the optimal sensor placement based on the simulation results. Afterwards, we apply this to a real robotic limb.

A. Optimal Sensor Placement and Validation in Simulation

Based on the preprocessed dataset acquired using finite element simulation (see Section II-A and II-B), we compare the performance of the sensor placement using the different selection methods.

In Fig. 5a we present the selected sensor positions. We notice that the PCA-QR method places the sensors on the edges of each beam whose deformation has the highest variances. In contrast, the SVR method arranges sensors onto the two off-center parallel beams ($y \approx 70\,\text{mm}$ and $y \approx 130\,\text{mm}$),

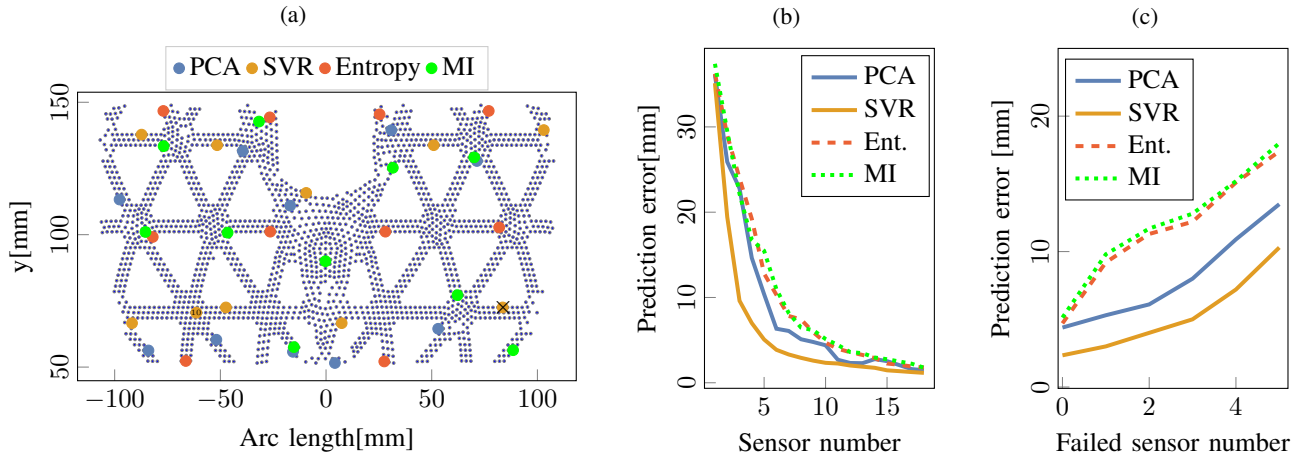

Fig. 5: Sensor placement results. (a) Selected optimal positions based on different criteria. × indicates a physical sensor with failure, see Section III-C. (b) Prediction error on test data set comparing different methods. (c) Robustness check for all methods.

in which deformation from both sides can be measured. The 10^{th} sensor position suggested by SVR is located closely to two already selected sensors. One would have expected a more centered position. As it turns out, also further sensors ($11^{th} - 30^{th}$ not shown) suggested by SVR are mostly at the boundaries instead of the center. We hypothesize that this is because the areas near boundaries are more rigid and less sensitive to applied force which requires a higher sensor density to get high prediction precision. The model-based methods (Entropy, MI) are only based on the geometric information. The Entropy criterion recommends the sensors to be placed toward the edges and homogeneously distributes them on the entire space. The Mutual Information criterion suggests the positions to be more centered.

After picking the optimal sensor positions, we evaluate the four methods by comparing the prediction performance using the SVR force location inference (Section II-C). As shown in Fig. 5b, data-driven methods work generally better because they can exploit the structure of the data. Note that the PCA-QR method is not a greedy method and it suggests different combinations of sensors for each sensor budget K. For that reason, the prediction error is not guaranteed to be in descending order.

In general, we notice that a small number of sensors can already lead to a relatively small prediction error (on unseen locations). Based on the simulation data we obtain a $< 10\,\mathrm{mm}$ precision for 5 sensors, selected using the greedy SVR method (Section II-D.1). With 10 sensors the prediction performance is roughly $2.5\,\mathrm{mm}$.

1) Probing Robustness to Failure: As any physical device is susceptible to failure, we also checked the robustness of the selected positions against failing sensors. We tested the prediction error with different degrees of sensor failures, i. e. out of 10 sensors 1 to 5 are broken. Fig. 5c presents the results. The greedy SVR method has the highest robustness against sensor failures. For $3/10$ failures the performance drops by $5\,\mathrm{mm}$.

B. Experimental Evaluation of Hardware

We use the 10 selected sensor positions by the greedy SVR method and implemented it on a hardware limb. The next subsections introduce the sensor choice, the sensor assembly process, the data acquisition and finally the results of the force inference on the real system.

1) Sensor choice: In this section, we motivate the choice of the physical sensors. We have chosen Strain Gauge (SG) sensors because they are generally cheap, widely available and relatively straightforward to use. Although we have also found cases of broken sensors after assembly. In order to work with the strong deformations of the 3D-printed plastic robot parts, we select SGs with 20% elongation rate. In our settings, the SG's finite extension is within the maximum elastic deformation of the limb. These have a long lifetime and high fatigue strength. One drawback of SG is that it only measures deformation along one direction. For our beam-shaped structure, the SGs are assembled along the beam directions where the dominant deformation happens according to FEM.

Alternatively, we considered active sensors such as piezo-electric sensors [22] transforming mechanical deformation into electrical energy and triboelectric effect based sensors [3] charging through frictional contact. A typical problem of these sensors is that they cannot detect static loads. In terms of passive sensors we considered capacitive ones, but they will react differently for conducting and consisting spatial deformation limits. Another interesting category is optical sensors like Fiber Bragg grating measuring a change in a deformed glass fiber based on the wavelength change of the reflected light. The processing equipment for these sensors is rather involved, expensive and bulky.

2) Sensor assembly: The selected Strain Gauge (SG) sensor has to be attached to the inside of the 3D plastic structure with precaution in order to avoid damage or malfunction. The assembly procedure is a bit challenging because it is inside the hollow object. We developed an assembly method with a specific support structure (SS), as shown in Fig. 6a. The

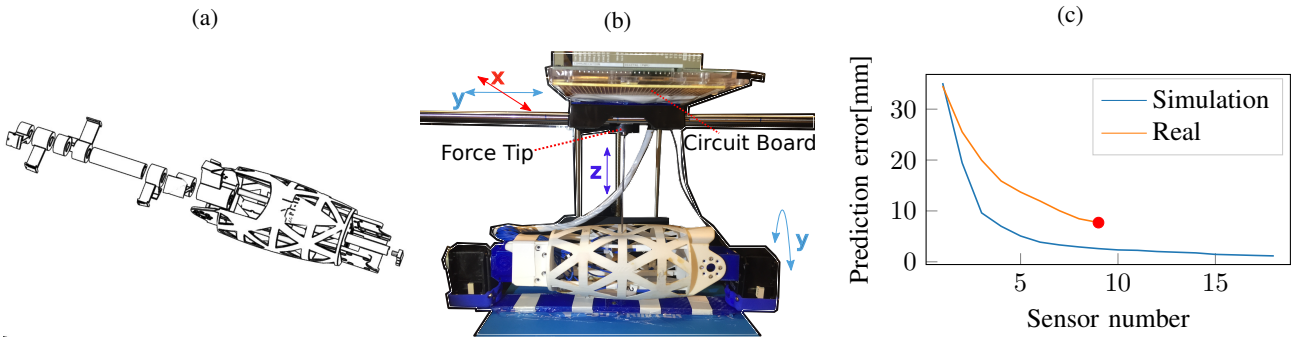

Fig. 6: Real world setup and results. (a) Support structure (SS) for assembling the strain gauges (SG). (b) Test-bed with 4 Degrees of Freedom to realize a sampling of single touch everywhere on the surface. (c) Prediction error comparison between simulation and real physical application.

SS has little arms to pre-tighten each of the sensors at the right position properly to the surface. The arms are held in place by a middle axis during the adhesive curing process and can be pulled out for disassembly. The assembly process is described in Tab. I.

TABLE I: Assembly procedure

Step	Details
1	Wire SG and cover SG with scotch tape to isolate adhesive.
2	Cover SS with preservative film to isolate adhesive.
3	Insert absorbent wool between SG and SS to absorb adhesive.
4	Position SG on SS.
5	Clean internal surface of skeleton and SG surface.
6	Coat SG and internal skeleton surface with prepared adhesive.
7	Pre-tighten the whole structure and cure for $6h$ at $+20°C$
8	Disassemble SS and clean surface.

3) Data Acquisition: To acquire the dataset for training the machine learning algorithm, we have to record the sensor measurements for many force applications.

a) Amplifier Circuit: The conventional data acquisition circuit for SG is the Wheatstone bridge [23] which measures electrical resistance's change by balancing two legs of a bridge circuit. As SGs are sensitive not only to mechanical stress and but also to temperature variance, temperature compensation function has to be integrated into the circuit.

In our project, we adopt half-bridge Wheatstone, operational amplifier of MCP609 and Arduino Due. The circuit has 12 I/O ports with 12 bits of resolution in which the SGs' deformation is amplified by a factor of 330 and converted to 4096 different values over $5\,\mathrm{V}$.

b) Test-bed: To collect a large amount of force-measurements in an automated way, we designed test-bed based on a modified 3D printer. The printer offers 3 Degrees of Freedom (DoF) given by the Cartesian translation in x, y, z. In addition, we add one axis of rotation. In order to measure the forces, the print-head of the printer is replaced by a force sensor tip (FC2231), see Fig. 6b.

C. Experimental Results

In this section, we validate the proposed HAPDEF system on the modified limb of the Poppy robot [1]. The limb is designed to have an inside rigid support and a flexible shell to detect the touch. The support structure sustains the forces acting at the joints and avoids shell deformations without touch. We assembled 10 sensors according to the placement determined in Fig. 5a. One of the sensors was malfunctioning, as indicated by a cross in the figure. Each sensor value is calibrated to be zero if no force is applied. While recording from the 9 remaining sensors, the force tip of the test-bed is stimulating the surface with different forces. More specifically, the force tip is moved towards the structure to apply nominal force. As soon as a contact is registered, the force tip is moved in small steps to a maximum penetration depth of $2\,\mathrm{mm}$. In this way, different force magnitudes are obtained per location. We collect data for 3000 locations on the surface, avoiding the edges and boundaries so that the force tip does not slide off.

From each location, we first use the largest force simulation and split the 3000 data points into 80% training and 10% for validation and test respectively. After training the Support Vector Regression and performing hyper-parameter selection based on the validation set ($C = 20$, $\epsilon = 10^{-6}$ and $\gamma = 2 \times 10^{-3}$), we evaluate the inference performance on the unseen force-locations. Fig. 6c presents the results compared to the simulations. The hardware implementation achieves half of the simulation accuracy. The average prediction precision of the force position is below $8\,\mathrm{mm}$ when using 9 sensors. Given that our structure has a total surface of $200\,\mathrm{mm} \times 120\,\mathrm{mm}$, this is a very high precision.

Besides, we train the SVR for different force amplitude, varying from light to strong touch, and report the prediction precision of the force information in different force intervals. As shown in Tab. II, SVR has low prediction precision for light touch and high precision for strong touch. Presumably because fewer sensors get activated by light touch. The absolute prediction precision for the force's amplitude varies little w. r. t. force strength. Consequently, any strong touches on the surface can be reliably detected and be used as a warning signal. This improves the haptic system's robustness.

IV. DISCUSSION

We present a method to obtain a robust haptic sensing system using only a small number of inexpensive deformation

TABLE II: Prediction precision of force's postion and amplitude w. r. t. different force strength

Force Interval	Position Error [mm]	Amplitude Error [N]
0 - 4.9 N	25.43 $\pm$ 13.36	1.05 $\pm$ 1.01
4.9 - 9.8 N	11.95 $\pm$ 11.85	1.19 $\pm$ 1.19
9.8 - 19.6 N	5.90 $\pm$ 7.79	1.42 $\pm$ 1.78
19.6 - 34.3 N	4.48 $\pm$ 6.29	1.54 $\pm$ 2.21

sensors. The performance of the sensation device is powered by a machine learning approach. After a learning period, the system can reliably localize touch all around a curved surface. Apart from being inexpensive, the system is also very durable as the deformation sensors can be placed inside the structure. Only a few (~ 10) sensor values need to be acquired and processed. The computational requirements are also low during operation as the inference of the force location is done via Support Vector Regression. However, other machine learning methods, such as Deep Neural Networks, are also feasible.

We also compared different methods of computing optimal sensor locations for a very sparse sensor configuration. We found that data-driven methods outperform geometry-based methods. The right selection strategy can reduce the required number of sensors by 50% without significant loss in precision. In future work, we want to investigate multi-touch and more accurate force information prediction.

ACKNOWLEDGMENT

The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) and the China Scholarship Council (CSC) for supporting Huanbo Sun. We would also like to thank Katherine Kuchenbecker and Jonathan Fiene for their valuable input on the project as well as Jia-Jie Zhu, Michal Rolinek, and Sebastian Bläs for the fruitful discussion.

REFERENCES

[1] M. Lapeyre, P. Rouanet, J. Grizou, S. Nguyen, F. Depraetre, A. Le Falher, and P.-Y. Oudeyer, "Poppy project: Open-source fabrication of 3d printed humanoid robot for science, education and art," in *Digital Intelligence 2014*, Nantes, France, Sept. 2014, p. 6. [Online]. Available: https://hal.inria.fr/hal-01096338

[2] G. H. Büscher, R. Kõiva, C. Schürmann, R. Haschke, and H. J. Ritter, "Flexible and stretchable fabric-based tactile sensor," *Robot. Auton. Syst.*, vol. 63, no. P3, pp. 244–252, Jan. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2014.09.007

[3] S. Wang, L. Lin, and Z. L. Wang, "Triboelectric nanogenerators as self-powered active sensors," *Nano Energy*, vol. 11, pp. 436 – 462, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2211285514002171

[4] M. Shimojo, A. Namiki, M. Ishikawa, R. Makino, and K. Mabuchi, "A tactile sensor sheet using pressure conductive rubber with electrical-wires stitched method," *Sensors Journal, IEEE*, vol. 4, no. 5, pp. 589–596, Oct. 2004.

[5] H. Lee, "Soft nanocomposite based multi-point, multi-directional strain mapping sensor using anisotropic electrical impedance tomography," *Scientific Reports*, vol. 7, no. 39837, Jan. 2017. [Online]. Available: http://doi.org/10.1038/srep39837

[6] J. A. Fishel and G. E. Loeb, "Sensing tactile microvibrations with the biotac - comparison with human sensitivity," in *Proc. IEEE-RAS and EMBS Int. Conf. Biomedical Robotics and Biomechatronics(BioRob)*. IEEE, 2012.

[7] N. Wettels and G. E. Loeb, "Haptic feature extraction from a biomimetic tactile sensor: Force, contact loaction and curvature," in *Proc. IEEE Int. Conf. Robotics and Biomimetics*. IEEE, 2011.

[8] B. Ward-Cherrier, N. Pestell, L. Cramphorn, B. Winstone, M. E. Giannaccini, J. Rossiter, and N. F. Lepora, "The tactip family: Soft optical tactile sensors with 3d-printed biomimetic morphologies," *Soft Robotics*, vol. 5, no. 2, pp. 216–227, 2018, pMID: 29297773. [Online]. Available: https://doi.org/10.1089/soro.2017.0052

[9] K. Lawrence, *ANSYS Tutorial Release 13*. SDC Publications, 2011.

[10] E. Candès and M. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[11] K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Data-driven sparse sensor placement," *CoRR*, vol. abs/1701.07569, 2017.

[12] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sept. 1995. [Online]. Available: https://doi.org/10.1023/A:1022627411411

[13] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, Aug. 2004. [Online]. Available: https://doi.org/10.1023/B:STCO.0000035301.49549.88

[14] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] E. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, 2008.

[16] I. Jolliffe, *Principal Component Analysis*. Springer Verlag, 1986.

[17] N. Cressie, *Statistics for spatial data*, ser. Wiley series in probability and mathematical statistics: Applied probability and statistics. J. Wiley, 1993. [Online]. Available: https://books.google.de/books?id=4SdRAAAAMAAJ

[18] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*. New York, NY, USA: Cambridge University Press, 2002.

[19] E. Ventsel and T. Krauthammer, *Thin Plates and Shells: Theory: Analysis, and Applications*. Taylor & Francis, 2001. [Online]. Available: https://books.google.de/books?id=-veAngEACAAJ

[20] C. E. Rasmussen, "Gaussian processes for machine learning." MIT Press, 2006.

[21] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research (JMLR)*, vol. 9, pp. 235–284, February 2008.

[22] T. H. Ng and W. H. Liao, "Sensitivity analysis and energy harvesting for a self-powered piezoelectric sensor," *Journal of Intelligent Material Systems and Structures*, vol. 16, no. 10, pp. 785–797, 2005. [Online]. Available: https://doi.org/10.1177/1045389X05053151

[23] D. M. Stefanescu, "Strain gauges and wheatstone bridges basic instrumentation and new applications for electrical measurement of non-electrical quantities," *Eighth International Multi-Conference on Systems, Signals and Devices*, pp. 1–5, 2011.