Human-Inspired Neurorobotic System for Classifying Surface Textures by Touch

Ken E. Friedl[1,4], Aaron R. Voelker[2,4], Angelika Peer[3], and Chris Eliasmith[2]

***Abstract*—Giving robots the ability to classify surface textures requires appropriate sensors and algorithms. Inspired by the biology of human tactile perception, we implement a neurorobotic texture classifier with a recurrent spiking neural network, using a novel semi-supervised approach for classifying dynamic stimuli. Input to the network is supplied by accelerometers mounted on a robotic arm. The sensor data is encoded by a heterogeneous population of neurons, modeled to match the spiking activity of mechanoreceptor cells. This activity is convolved by a hidden layer using bandpass filters to extract nonlinear frequency information from the spike trains. The resulting high-dimensional feature representation is then continuously classified using a neurally implemented support vector machine. We demonstrate that our system classifies 18 metal surface textures scanned in two opposite directions at a constant velocity. We also demonstrate that our approach significantly improves upon a baseline model that does not use the described feature extraction. This method can be performed in real-time using neuromorphic hardware, and can be extended to other applications that process dynamic stimuli online.**

***Index Terms*—neurorobotics, biologically-inspired robots, force and tactile sensing**

I. Introduction

HUMANS are remarkably adept at perceiving the environment using their sense of touch. By moving the fingertip across a surface, vibrations give rise to perceptual qualities such as roughness [1], stickiness, and slipperiness [2], [3]. Thus, structural details smaller than $1\,\mu\mathrm{m}$ in different quality grades of silk, paper, and grind metal surfaces can be differentiated [4]. Our goal is to draw inspiration from nature to give robots the ability to classify surface textures in real-time. This ability can be deployed within systems that need to classify tactile stimuli, such as those employed to automate quality control of textured surfaces.

Dynamic robotic systems for tactile surface sensing have been developed using various technologies [5]. In this context, *tactile* implies the sensors physically touch the textured surface. *Dynamic* refers to robotic setups that perform exploratory movements across the surface to accumulate sensor data. Accelerometers have been used as dynamic vibration sensors for texture discrimination in [6], [7] and [8]. By using only the signal variance of two spatially separated vibration sensors, four different textures could be differentiated in [9]. In addition to variance, Giguere and Dudek [7] utilized mean, skewness, kurtosis and some higher order moments of one accelerometer to differentiate between ten fine and coarse textures. Other studies have used multiple exploratory movements to boost classification accuracy. For instance, Sinapov et al. [8] used five exploratory movements with varying direction and speed, and analyzed the spectrotemporal features to classify twenty naturalistic fine textures with 80% accuracy. Another study kept executing exploratory movements until at least 80% of the movements indicated a specific texture [10].

However, only a few groups have developed classifiers using spiking neural networks. A closed perception-action loop was built to classify Braille characters [11], by supplying pressure sensor array data to *leaky integrate-and-fire* (LIF) neurons, and then using a naive Bayes classifier to control the scanning velocity. Another approach classified ten naturalistic textures with 97% accuracy [12], by simulating an Izhikevich neuron in response to an array of four piezoresistive sensors, and analyzing the precise spike timing using spike distance metrics. However, these metrics require offline processing of the entire time window, and therefore cannot be implemented in real-time.

Related work has also used *Bayesian exploration* to optimize the information supporting various textural properties [13]. This model is capable of using a number of exploratory movements to classify 117 textures with a 95.4% success rate. However, the system requires an offline phase to reduce the input signal to a number of scalar values, making it unsuitable for real-time applications. This reduction could also discard valuable information from the input signal, since it is manually crafted using a finite set of properties motivated by psychophysical studies.

In contrast, we find it possible to automatically discover the most salient *features* from a high-dimensional representation of the input stimulus, and classify these features *online*. This is challenging since the relationship between a rapidly time-varying tactile stimulus and its resulting perceptual qualities is highly nonlinear [4]. To make this problem tractable, we translate our biological understanding of tactile perception into an artificial network of spiking neurons, and then train the desired function. Our approach is implemented entirely within

Manuscript received: August, 31, 2015; Revised November, 18, 2015; Accepted December, 22, 2015.

This paper was recommended for publication by Wan Kyun Chung upon evaluation of the Associate Editor and Reviewers' comments.

[1]Chair of Automatic Control Engineering, Technische Universität München, D-80333 Munich, Germany, Email: friedl@tum.de

[2]Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, ON, Canada N2L 3G1, Email: arvoelke@uwaterloo.ca

[3]Bristol Robotics Laboratory, University of the West of England, T Block, Frenchay Campus, Coldharbour Lane, Bristol, BS16 1QY, United Kingdom, Email: angelika.peer@brl.ac.uk

[4]Aaron R. Voelker and Ken E. Friedl contributed equally to this work. The idea for this work, theoretical background for the system and experimental setup was done by Ken E. Friedl. The code, architecture, and theory were developed by Aaron R. Voelker.

Digital Object Identifier xxxxxxxxxxxxxxxxxxx

a spiking neural network that can be simulated on low-power neuromorphic hardware.

When fingers move across a texture, tiny hills and valleys along its surface rapidly excite various types of static and dynamic *mechanoreceptors* [4]. Biomechanical processes within these cells convert the physical forces applied by the stimulus into spike trains. We adapt a mechanoreceptor model by Kim et al. [14] to simulate this process in a population of spiking neurons. The nervous system routes the mechanoreceptor responses through the spinal cord where they are processed by cuneate neurons [15], which function as feature extractors across this spiking activity [16]. This motivates the inclusion of a *hidden layer* in the model to extract a high-dimensional feature representation. We have designed this layer to encode frequency bands of the stimulus, since frequency is an important factor in psychophysical experiments involving the differentiation of textures [1]. The population of cuneate neurons project these features to somatosensory cortex, where an abstract percept of the texture is thought to be formed [16]. Our model learns the mapping from features to textures using a support vector machine (SVM). These learned weights are embedded within the connection to a recurrent output layer. To summarize, the network encodes the physical stimulus into spiking activity, maps this activity to extract salient features, and finally decodes the resulting spikes to obtain an online classification.

To implement this model, we employ the Neural Engineering Framework (NEF), which provides a method for mapping functions to a biologically plausible spiking neural network [17]. These networks can be efficiently simulated on neuromorphic hardware such as SpiNNaker [18], resulting in significant energy savings [19]. Simulations are also more efficient than all-to-all connected networks due to the use of factored weight matrices [20], and are generally robust to noise and physical variability due to heterogeneity [21].

Our method uses the NEF to classify surface textures scanned with a robotic setup. We report performance on a set of 18 metal surface textures. This establishes a novel biologically-inspired system that can be simulated in real-time using neuromorphic hardware.

II. Materials & Methods

A. Neural Network Architecture

1) Neural Engineering Framework: Our approach uses the NEF, which consists of three principles that enable the translation of a high-level mathematical description of a system into a biologically plausible model of spiking neurons and connection weight matrices [17].

The first principle provides a way of representing a time-varying vector $\mathbf{x}(t)$ in the spiking activity of a population of neurons. In particular, the spiking activity a_i at time t for the i^{th} neuron in the population is given by:

$$a_i(t) = G[\alpha_i \mathbf{e}_i \cdot (\mathbf{x} * h_i)(t) + \beta_i], \tag{1}$$

where $\cdot$ denotes a dot product and $*$ convolution. Here, each neuron i has an *encoding vector* $\mathbf{e}_i$ which can be understood as the vector for which the neuron will fire most strongly. The input current to the neural nonlinearity $G[\cdot]$ at time t is a linear function of $\mathbf{x}(t)$, with coefficients given by $\mathbf{e}_i$, scaled by a gain $\alpha_i > 0$, plus a bias current β_i. In addition, there is a causal linear filter[1] $h_i(t)$ that models the post-synaptic current (PSC) resulting from the arrival of a spike at the synapse of the i^{th} neuron. We note that this principle decouples the number of neurons that represent $\mathbf{x}(t)$, from the dimensionality of this vector.

The neural nonlinearity $G[\cdot]$, may in theory be any model that converts a time-series of current into a spike train. This work simulates both adaptive and non-adaptive leaky integrate-and-fire (LIF) neurons to generate spikes, since they offer an ideal trade-off between efficiency and biological plausibility [17]. LIF neurons operate by simulating a resistor-capacitor circuit that maps on to membrane properties of neurons. If the membrane potential passes a threshold, the neuron emits a spike, resets for an absolute refractory period (2 ms here), and begins to respond to the input voltage again. Similarly, adaptation was added to the LIF model by reducing the input current by an adaptive term that increases every time the neuron spikes, and decays over time [22].

To recover the vector from the population's resulting activity, each neuron has a *decoding vector* $\mathbf{d}_i$, such that

$$\hat{\mathbf{x}}(t) = \sum_i (a_i(t) + \mu)\mathbf{d}_i, \tag{2}$$

where $\mu \sim \mathcal{N}(0, \sigma^2)$ models Gaussian noise in the neural activity. These decoding vectors are found by optimizing the root mean squared error (RMSE) of $\hat{\mathbf{x}}(t) - \mathbf{x}(t)$ using a standard regularized least squares solver.

The second principle observes that to compute the identity function between two populations of neurons, the connection weights W can be factored into a matrix of encoders $\mathbf{e}$ for the post-synaptic neurons and the decoders $\mathbf{d}$ from the pre-synaptic neurons.

$$W = \mathbf{e}\mathbf{d}^T \tag{3}$$

Due to the neural nonlinearity $G[\cdot]$, the same equation can also be used to approximate any nonlinear transformation, by modifying (2) to decode $\hat{f}(\mathbf{x}(t)) = \sum_i (a_i(t) + \mu)\mathbf{d}_i^f$, where $\mathbf{d}_i^f$ are the optimal linear decoders for approximating the function f, by minimizing the RMSE of $\hat{f}(\mathbf{x}(t)) - f(\mathbf{x}(t))$.

The third principle provides a method of mapping a control-theoretic description of a dynamical system into a recurrently connected population. For the purposes of this work, we are mainly interested in the simple case of a *leaky integrator*, i.e. lowpass filter with a desired time constant τ_0:

$$\dot{\mathbf{x}}(\mathbf{t}) = \frac{1}{\tau_0}(\mathbf{u}(t) - \mathbf{x}(t)). \tag{4}$$

By using the standard lowpass PSC filter with time constant τ, it can be shown that $\mathbf{x}(t)$ obeys (4) when

$$A' = 1 - \frac{\tau}{\tau_0}, \quad B' = \frac{\tau}{\tau_0}, \tag{5}$$

[1]This is typically the standard exponential model $h_i(t) = (1/\tau)e^{-t/\tau}$, which performs lowpass filtering with time constant τ. This work also utilizes highpass and bandpass filters that differ per synapse.

where A' is a scalar multiplying the recurrent decoding vectors, and B' is a scalar multiplying the input $\mathbf{u}(t)$ [17].

The neural architecture of our system uses these principles of the NEF at each of its three layers: a population of mechanoreceptors, a hidden layer of features, and a recurrently connected output layer (see Fig. 1). We proceed by describing each of these layers individually.

2) Mechanoreceptor Population: The human sense of touch uses free nerve endings and specialized mechanoreceptors to sense vibrations on the skin. The latter are composed of two types: 1) slowly adapting type cells which are sensitive to static stimuli; 2) the so-called rapidly adapting type cells. The slowly adapting cells include Merkel (SA1, static pressure) and Ruffini cells (SA2, skin stretch). The rapidly adapting cells are sensitive to transient events such as vibration. These include Meisser (RA, vibration in the range of 1 to about 60 Hz) and Pacinian cells (PC, vibrations in the range of 30 to about 700 Hz) [23]. When the finger moves across a surface texture, the skin is excited by vibration patterns influenced by macro- and microscopic topography of the surface in contact. These cells transform the physical stimulus into spiking patterns.

To mimic this biological representation, our model encodes the vibrations recorded from multiple accelerometers into the spiking activity of a heterogeneous population of mechanoreceptor neuron models. Since SA2 cells do not contribute directly to fine textural percepts [4], we only consider PC, RA, and SA1 type cells for our model.

The mechanoreceptor model by Kim et al. [14] has been shown to accurately reproduce the spike trains of RA and SA1 type cells on a variety of stimuli. Following this model, we take the acceleration $u(t)$, and its two derivatives $\dot{u}(t)$, $\ddot{u}(t)$, and separate each of them into positive and negative rectified parts resulting in 6 signals. Since acceleration is directly proportional to the net force, we interpret $u(t)$ as the force applied to the fingertip. Then $\dot{u}(t)$ and $\ddot{u}(t)$ are the first- and second-order changes in force, respectively. As shown in Fig. 1, the rectified signals form 6 inputs, which are weighted and summed to form the current to an adaptive LIF neuron.

This is equivalent to the neural representation of a 6-dimensional vector $\mathbf{x}(t)$ in the NEF using (1). Each encoding vector $\mathbf{e}_i$ corresponds to the 6 weights that depend on the cell type of the i^{th} neuron. Different weights then reproduce the spiking characteristics of different types of mechanoreceptors. For the PSC model required by the NEF, we use a lowpass filter with time constant $\tau = 1$ ms on all mechanoreceptors. We found experimentally that this constant works well to balance the need for smoothing (large τ) with the need to maintain high frequencies in the signal (small τ). To simplify implementation, differentiation is performed by combining this PSC filter with a highpass filter with the same time constant (Laplace transform τs). Our model differs from the original model only by the use of an adaptive LIF model instead of the *Mihalas-Niebur* model, and by omitting a saturation filter applied to the current. The adaptive LIF model is used in place of $G[\cdot]$ to obtain a neural model that is simpler than Mihalas-Niebur while still supporting adaptation. We then use (1) to simulate the spiking activity of each neuron in response to the stimulus.

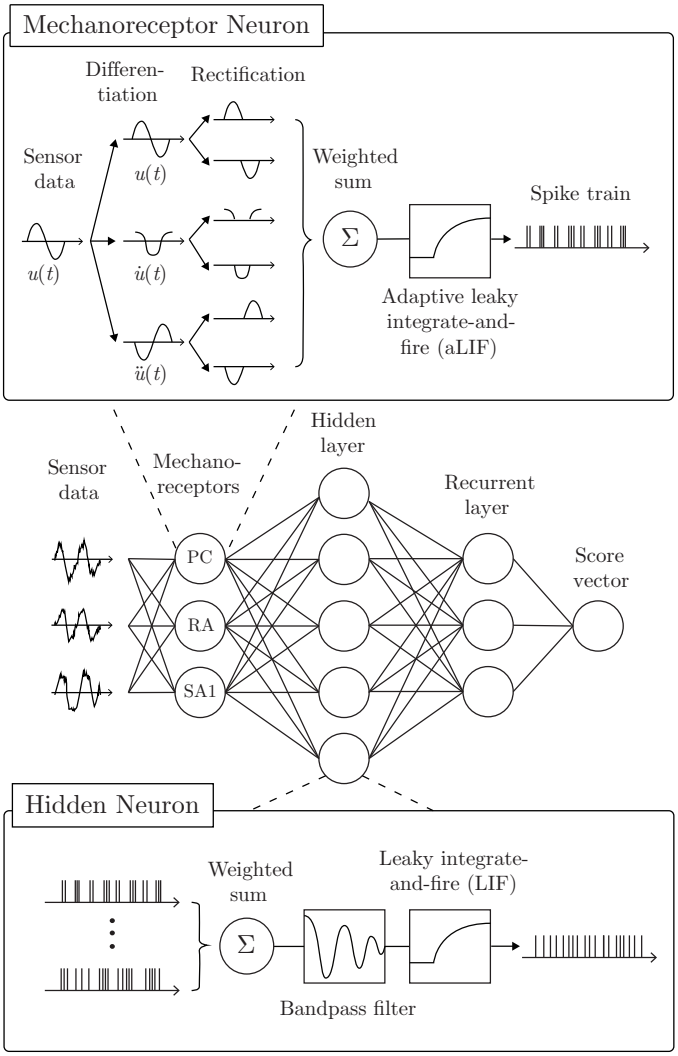

Fig. 1. Architecture of the surface texture classifier. Data from three sensors are encoded in the spiking activity of mechanoreceptor neurons. A hidden layer extracts nonlinear features of these spike trains in the frequency domain by convolution with bandpass filters. A recurrently connected output (feedback arrows omitted) represents the score for each class, smoothed over time. This score vector is linearly decoded from the recurrent layer to find the texture with the largest score. Connection weights and neural parameters are given by the principles of the NEF.

Rather than fitting the parameters of the neurons directly to individual model neurons as in the original model, we uniformly randomize parameters α_i, β_i, and $\mathbf{e}_i$ of 40,000 neurons (this number was chosen to sufficiently sample the space of parameters), and then select those with the highest Γ-*factor* (GF) [24] to canonical PC/RA/SA1 models on a standard test stimulus[2]. The GF compares the number of coincidental spikes within a ± 2 ms window to the expected value generated by a Poisson process with the same firing rate: 1 implies a perfect match; 0 a Poisson process; and -1 indicates anti-correlation. We use this particular measure to be consistent with the prior analysis of the original model. The pruned population of mechanoreceptors contains $20/300/200$

[2]Canonical models and test data were obtained by email correspondence with the authors of [14]. Although PC type cells were not evaluated by the original study, their model included sample PC parameters.

neurons (520 total) of type PC/RA/SA1, respectively, to match the distribution within a $2\,\text{cm}^2$ surface of contact [23]. The GF of selected neurons range from 0.19 to 0.99 with mean 0.49 across all three types. This is promising considering Kim et al. [14] found the mean GF between individually fit models and their canonical RA/SA1 models to be approximately 0.3.

The resulting distribution of encoders effectively reflects the sensitivity of each cell to various characteristics of the sensory input. By allowing them to vary randomly, the information content of the population is increased [21]. In particular, the network can further process the 6-dimensional representation using (2) while being robust to noise. This need to exploit individual variability is consistent with the conclusion made by Kim et al. [14] that heterogeneity in afferent fibers matters when conveying precise timing information about the tactile response.

The fingertip has multiple points of contact while scanning a surface, allowing it to pick up spatial features of the stimulus [4]. The above 6-dimensional encoding is repeated 3 times (for each sensor in our robotic setup) to form an 18-dimensional representation, where the encoding vector for each neuron is tuned to different combinations of dimensions from the different sensors. This further improves robustness to noise by exploiting multiple correlated sources of information. The distribution of encoders could also be controlled to match known spatial characteristics of each cell, but for simplicity we set all encoders to be uniformly distributed across the 3 sensor signals.

3) Hidden Layer: Feature Extraction in the Frequency Domain: Given the spiking activity of the population of mechanoreceptors, we now extract a high-dimensional representation in a hidden layer of neurons, to capture general features of the stimulus.

The 18-dimensional input representation (6 dimensions $\times$ 3 sensors) is recovered from the spiking activity of the mechanoreceptor population, by solving (2) for a 520×18 decoder matrix $\mathbf{d}$ (18 input dimensions for each of the 520 mechanoreceptor neurons) using regularized least squares optimization. The parameters of 7,200 LIF neurons (this number was chosen to sufficiently sample the space of neural parameters per dimension), one for each feature in the hidden layer, are randomly generated, with encoders constrained to each of the possible 18 dimensions. This forms a sparse $7{,}200 \times 18$ encoder matrix $\mathbf{e}$ such that (3) gives the connection weight matrix from the mechanoreceptor population to the hidden layer. The hidden layer then represents the same 18-dimensional input at the current point in time, by having learned the optimal decoding from the spikes of the adaptive mechanoreceptor layer.

Motivated by the importance of stimulus frequency in psychophysical experiments involving texture discrimination [1], our aim is to extract features that encode information about the frequency of the current stimulus. The weight matrix transforms the activity independently of time, and is therefore insufficient to extract information about its frequency content. Thus, we provide the i^{th} neuron in the hidden layer with its own PSC filter $h_i(t)$, modeled as a 2^{nd} *order bandpass* filter with Laplace transform

$$H_i(s) = \frac{1}{\frac{1}{\omega_i^2}s^2 + \frac{1}{\omega_i Q_i}s + 1} \tag{6}$$

where ω_i is the peak frequency in radians per second, and Q_i is inversely proportional to the bandwidth. This filter was chosen because it is the linear filter of lowest order that is capable of isolating particular bands of frequencies.

Each ω_i is chosen randomly without examining the input data. Since differentiation is a form of highpass filtering, we expect the higher derivatives to represent higher frequency bands more accurately. Thus ω_i is chosen between $0-250$, $125-375$, or $250-500\,\text{Hz}$, depending on whether the neuron is encoding a dimension from $u(t)$, $\dot{u}(t)$, or $\ddot{u}(t)$, respectively. Frequencies above $500\,\text{Hz}$ are unnecessary due to the refractory rates of neurons ($2\,\text{ms}$). We found that randomizing Q_i between $2-50$, regardless of dimension, lead to the best results.

The spiking activity of each hidden neuron encodes a recovered input dimension convolved with a random bandpass filter (see Fig. 1). Decoding the square of a particular filter from this population then yields the power of that filter. More generally, any nonlinear functions of the frequency bands that are supported by the neural basis functions[3] can be used as a feature – a fact readily exploited in the following subsection.

4) Recurrent Layer: Classification of Features in a Neural Support Vector Machine: The final layer of our model is a recurrent output layer, which classifies the spiking activity of the hidden layer into one of k possible surface textures. The output layer is a population of 900 LIF neurons that represent a k-dimensional vector encoding a *score* for each class. The value of 900 was chosen to be large enough to accurately decode all k dimensions using (2).

Since the activity of the hidden layer is noisy and high-dimensional, we use scikit-learn (v0.16.1) to train a *one-vs-all linear support vector machine* (SVM), from the spikes of the hidden layer, without being prone to overfitting [25]. Since each of the k classifiers computes a dot product, together they form a $7{,}200 \times k$ decoder matrix $\mathbf{d}$ (one decoding vector for each neuron in the hidden layer) equivalent to solving (2) using the SVM maximum margin objective.

A $900 \times k$ encoder matrix $\mathbf{e}$ (one encoding vector for each neuron in the output layer) is randomly generated, and (3) provides the weight matrix to the output layer. To determine the resulting classification, we again solve for a $900 \times k$ decoder matrix $\mathbf{d}$, so that (2) produces an estimate of the k-dimensional score vector. The index of the maximum value yields the classification for the given point in time.

The population is recurrently connected to itself using (5) to integrate the scores with a leak term. This smooths the classification over time, with different recurrent lowpass time constants (τ_0) altering the rate of adaptation to new stimuli or the forgetting of old evidence. We use $\tau_0 = 50\,\text{ms}$ and $\tau = 5\,\text{ms}$ for the output layer. The value of τ matches the default synaptic time-constant in our simulation software,

[3]Eliasmith and Anderson [17] characterize this in detail using singular value decomposition on the activity matrix, and show that squaring is accurately supported by LIF tuning curves.

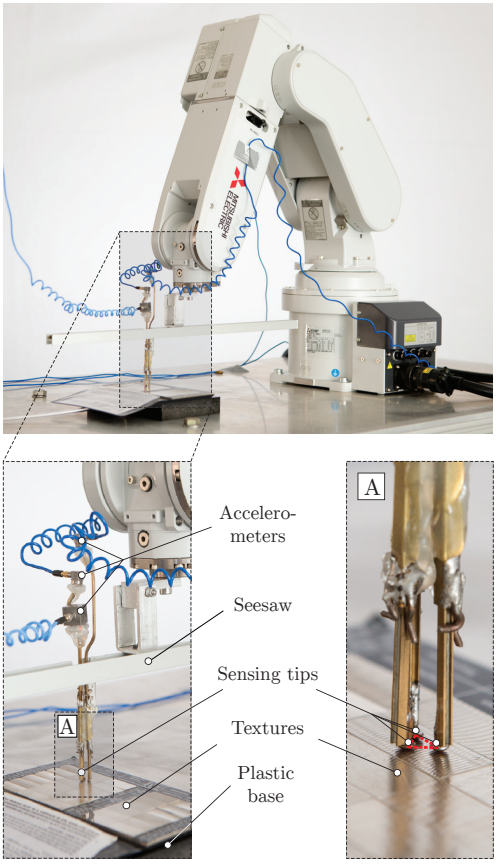

Fig. 2. Robotic setup using a 6 degree-of-freedom (6DoF) robotic arm and 3 sensors. (A) Triangular arrangement of the 3 sensing tips.

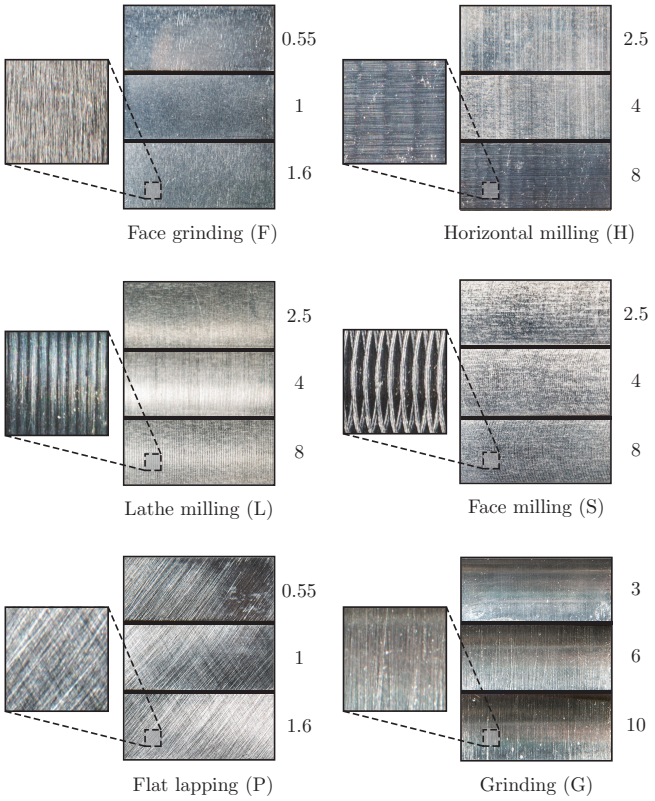

Fig. 3. Detailed photographs of surface textures used to test the system. This set was manufactured using 6 different metal surface finishing methods, with 3 roughnesses per finishing method, for a total of 18 distinct textures. The mean roughness depth (Rz) is indicated to the right of each texture.

while τ_0 was found to help persist correct classifications over time intervals with insufficient evidence in the signal.

It is important to note that although the SVM itself is linear, its features are not, due to the neural nonlinearities in the hidden layer. This is mathematically equivalent to using a kernel function given by the LIF tuning curves. Thus, the SVM nonlinearly separates the encoded frequency bands using randomly generated neural basis functions.

5) Nengo: All experiments were run using Nengo (v2.0.1), a Python implementation of the NEF [20]. Neurons were configured with default parameters, and the parameters from (1) were randomly sampled across the default ranges. Nengo automatically learned the decoders and connection weights offline, by optimizing the RMSE of (2) given randomly sampled $\mathbf{x}(t)$ from the unit hypersphere.

The simulations of (1) used a timestep of 0.5 ms (2 kHz). The accelerometer data $u(t)$ was streamed as input to the neural network. The spike train of the output population was read to obtain a classification at each timestep.

B. Robotic Setup

The robotic setup responsible for obtaining the sensor data consists of acceleration sensors mounted on a robotic arm and a sheet of textures (see Fig. 2).

1) Sensors: Three piezoceramic acceleration sensors, two one-axis PCB Piezoelectronics C65 and one three-axis 356A32 with only 1 DoF in use, are arranged in 120° triangular fashion with 5 mm spacing (see Fig. 2A). This arrangement is used to give our system the ability to pick up spatial features of the stimulus. Contact to the textured surfaces is achieved by mounting each sensor on a brass beam that is loosely supported in a brass tube. On the contact point to the texture, a copper tip is soldered to the beam to achieve a relatively soft contact preventing damage to the texture. A stable and vibrationless contact with 30 g load is achieved by a seesaw that is equipped with two brass weights. Sensors are electrically insulated from metal structures of the robotic arm and from one another. The analog signals are conditioned and amplified using PCB Piezoelectronics 482C Series Sensor signal conditioner. These signals are digitally sampled at 10 kHz using a DAQ card, National Instruments NI USB-6218, that is connected via USB to a computer running Windows 7 and controlled via MATLAB 2014a.

2) Robotic Arm: The robotic arm is a 6 degree-of-freedom manipulator, Mitsubishi Electric MELFA RV-2F-D. A MELFA CR750-D controller receives position commands in task space provided through UDP with a custom C++ program to maintain constant velocity.

3) Textures: The data set consists of 18 textures manufactured by the following 6 metal surface finishing methods: face grinding (F), horizontal milling (H), lathe milling (L), face milling (S), flat lapping (P), and grinding (G), with three roughnesses per finishing method (see Fig. 3). These

textures were selected to include a broad range of geometric dimensions of fine texture detail, with mean roughness depth (Rz) ranging from extremely fine ($0.55\,\mu$m) to relatively coarse ($10\,\mu$m). The sheet of textures was mechanically insulated from the table to which the robot was fixed and then glued to a plastic base ($100\,$mm $\times$ $150\,$mm $\times$ $20\,$mm).

4) Experimental Protocol: Before each data acquisition session, the load on the sensors was measured using an electronic scale. Prior to placing the textures, the sensing triangle was lowered to the surface of the scale using the robotic arm. To ensure constant pressure, the two brass weights were moved back and forth on the seesaw until a stable load of $30\,$g was obtained. The scale was then replaced by the plastic base holding the textures. Air was supplied beneath the plastic base at $30\,$psi to suspend the textures above a thin film of air. The center of the sensing triangle was set to start vertically centered at $11\,$mm from the top of the texture, and $6\,$mm from the side.

Each texture was scanned parallel to its longest edge, at a constant velocity of $0.7\,$mm/s for $120\,$s. After moving $10\,$mm, the robot would stop for $0.5\,$s (while the recording continues) and then repeat in the opposite direction. Although different scanning velocities were not considered by this study, our current experimental setup and protocol is robust and generally applicable.

To determine the scanning velocity, we measured the velocity range of three human subjects (1 female aged 25, 2 males aged 26 and 28) during lateral movements along the longest edge of each texture, which yielded a range of velocities between $0.7\,$mm/s and $23\,$mm/s. This was accomplished with a tracking system (Qualisys), by attaching an infrared marker to each participant's index finger during an identification task involving three of the textures. We chose the lowest velocity of $0.7\,$mm/s, since this was observed to minimize background vibrations from the robotic arm.

All sensor data was offset and scaled so that 99% fell within the interval $[-1, 1]$ with mean 0. To cross-validate the model, we randomly split the data into four folds, and validated the system four times (each time using $90\,$s of training data and $30\,$s of testing data per texture). Furthermore, each test signal was partitioned into thirty disjoint $1\,$s segments for classification (for a total of 120 examples per texture).

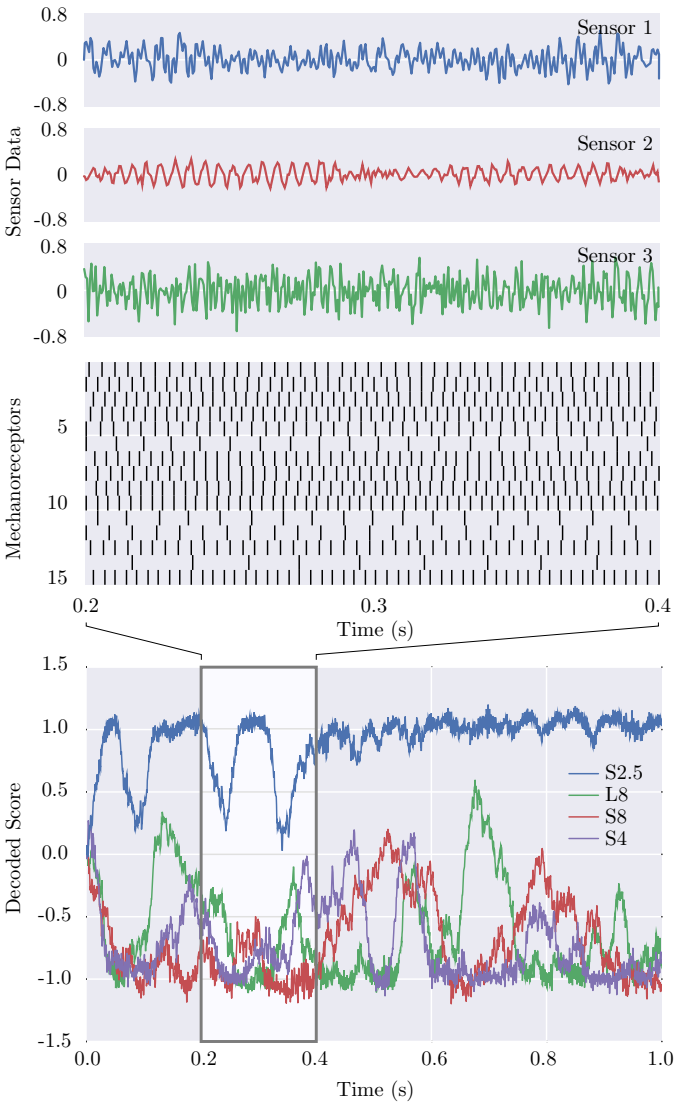

Fig. 4. Network activity on a 1 second test segment from the S2.5 recording. (Top) Normalized accelerometer data from the three sensors, across an interval of 0.2 to 0.4 s. (Middle) Spike raster of 15 (out of 520) randomly selected mechanoreceptor neurons, over this same time interval. (Bottom) Top 4 scores across the entire 1 s segment, obtained by decoding the output population. The network correctly classifies the texture as S2.5 at every timestep after the first 10 ms.

III. Results

To evaluate the system, the trained network was tested on each $1\,$s of test data. The 18-dimensional score vector was decoded from the spiking activity of the output population, lowpass filtered with default time constant ($5\,$ms), and sampled every $1\,$ms (see Fig. 4). A total score for each texture was obtained by summing together the individual score vectors across the test segment. The classification for each test segment was taken to be the texture with the highest total score. These results are broken down by texture in Table I, averaged across each test segment from all four folds, with an overall accuracy of 65.6%. The most common classification is the correct texture, in all cases.

By analyzing the trained network, we may succinctly characterize what each classifier is sensitive to in the frequency domain. We visualize this to indicate which bandpass parameters are most important for classifying all textures (see Fig. 5 top), and which frequencies are most important for classifying a specific texture (see Fig. 5 bottom). The top figure reveals that narrow filters (higher values of Q) tend to carry more weight, while frequencies in the range of $50-100$, $240-260$, and $400-500\,$Hz are less useful. The bottom figure shows for example that neurons encoding the negatively rectified dimension of $\dot{u}(t)$ will provide the most evidence for L2.5 when this dimension has power at $200\,$Hz.

We also compared our approach to a simpler model, by training and evaluating the same model without a hidden layer. This baseline model was prepared and validated under the same conditions, except the SVM used the spiking activity of the mechanoreceptor population as its features, rather than

TABLE I
CONFUSION MATRIX FOR 1 SECOND TEST SEGMENTS (%)

		Predicted																	
		F.55	F1	F1.6	H2.5	H4	H8	L2.5	L4	L8	S2.5	S4	S8	P.55	P1	P1.6	G3	G6	G10
Actual	F.55	**96.7**	0.8	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0	0.8
	F1	14.2	**55.8**	8.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	9.2	0.0	4.2	1.7	5.8
	F1.6	4.2	0.0	**80.0**	0.0	0.0	0.0	0.8	2.5	1.7	0.0	1.7	0.0	0.0	7.5	0.0	0.8	0.0	0.8
	H2.5	0.8	0.0	1.7	**40.8**	0.0	5.8	0.0	0.0	0.8	5.0	33.3	1.7	8.3	0.0	0.8	0.0	0.0	0.8
	H4	0.0	0.0	0.0	0.0	**99.2**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0
	H8	0.0	0.0	0.0	0.0	1.7	**68.3**	0.0	0.0	1.7	1.7	9.2	1.7	10.0	1.7	0.0	0.0	0.8	3.3
	L2.5	4.2	0.0	4.2	0.0	0.0	0.0	**73.3**	0.8	7.5	6.7	2.5	0.0	0.0	0.0	0.0	0.8	0.0	0.0
	L4	0.0	0.0	10.0	0.0	0.0	0.0	2.5	**60.0**	0.0	0.0	25.0	0.8	0.8	0.0	0.8	0.0	0.0	0.0
	L8	3.3	0.0	10.8	0.0	0.0	0.8	16.7	8.3	**49.2**	0.0	1.7	0.8	3.3	0.8	0.0	2.5	0.0	1.7
	S2.5	0.0	0.0	0.0	0.8	0.0	10.8	5.0	0.0	2.5	**67.5**	0.8	0.0	5.0	0.0	0.0	6.7	0.8	0.0
	S4	0.0	0.0	0.8	0.0	0.0	0.0	0.0	1.7	0.8	1.7	**92.5**	0.0	0.0	0.8	1.7	0.0	0.0	0.0
	S8	0.0	0.8	0.8	0.8	0.0	9.2	1.7	0.0	0.8	14.2	17.5	**40.0**	0.8	1.7	6.7	0.0	0.0	5.0
	P.55	0.8	0.0	0.8	0.8	0.0	7.5	0.0	0.0	0.8	0.0	2.5	0.0	**75.0**	5.8	0.0	4.2	0.0	1.7
	P1	0.8	0.0	1.7	0.0	0.0	5.0	0.0	0.8	0.0	0.0	2.5	0.0	10.8	**55.8**	0.0	6.7	0.8	15.0
	P1.6	0.0	0.0	0.0	0.8	0.0	0.8	0.0	0.8	0.8	0.0	0.8	0.0	0.0	0.0	**95.8**	0.0	0.0	0.0
	G3	10.0	0.0	2.5	0.0	0.0	2.5	0.0	0.0	0.0	17.5	0.0	0.0	4.2	7.5	1.7	**41.7**	7.5	5.0
	G6	10.0	6.7	2.5	0.0	0.8	4.2	0.8	0.0	0.0	1.7	3.3	0.0	1.7	5.8	2.5	25.8	**29.2**	5.0
	G10	0.0	0.0	10.0	0.0	1.7	0.8	1.7	0.0	2.5	0.0	3.3	0.0	0.8	18.3	0.0	0.8	0.0	**60.0**

the spiking activity from the hidden layer. Cross-validation accuracy decreased to 17.8%, averaged across each test segment from all four folds. We remark that the baseline still captures temporal information through its various lowpass filters (in the output layer and PSCs) and highpass filters (in the mechanoreceptor cells), yet it is no longer able to isolate particular bands of frequencies. Therefore, it is the addition of bandpass filters in the hidden layer that enable the network to accurately separate the feature space by texture.

IV. DISCUSSION & FUTURE WORK

We trained a three-layer network of spiking neurons to classify a set of 18 textures. A biologically plausible model of mechanoreceptors was adapted to encode the input vibrations. Psychophysical experiments and the role of cuneate neurons motivated a hidden layer that extracts frequency information. Lastly, an SVM determined the connection weights into a recurrently connected population. To our knowledge this is a novel semi-supervised approach for classifying dynamic stimuli using a spiking neural network.

A key advantage of our approach is that the network can be simulated in real-time using low-power neuromorphic hardware. At the same time, the NEF endows our model with benefits such as robustness to noise and parallel computation. Similarly, the system immediately provides classifications online, and performs well with brief inputs lasting only 1 s. These advantages make our method generally suitable for use in robotic applications, thus advancing the state of the art for texture classification.

A comparison with a baseline model revealed that performance was a consequence of the hidden layer. The first two layers of the network are unsupervised, and so the activity of the hidden layer represents general features that can in theory be reused for other applications. We intend to demonstrate this by extending our system to differentiate between textures, by interpreting the feature vector as a high-dimensional description of a texture. We also suspect that other tasks which process tactile stimuli can benefit by using this same vector.

Likewise, features of the input stimulus are learned and classified using general techniques from signal processing and machine learning. The methodology that we have described here need not be limited to the use of mechanoreceptor models and bandpass filters. While these tools were needed to appropriately constrain our model, other applications involving the processing of dynamic stimuli (e.g. visual or auditory) may readily place different constraints on how each layer encodes and filters information. This in turn may allow the architecture to be modified and redeployed within other domains.

The test results indicate how often each texture is confused with another. In general, it should be possible to design a simple psychophysical experiment to compare our system to human performance. However, our system is at a fundamental disadvantage since it does not alter its position or pressure to gather more evidence when unsure of its prediction. We are considering future extensions that solve this issue with a closed-loop system that can actively control its motor movements, with a range of velocities, based on feedback from the accumulated features.

ACKNOWLEDGMENT

This project was supported by the European Commission seventh framework programme under grant 610902, CFI, OIT, Canada Research Chairs NSERC Discovery grant 261453, ONR grant N000141310419, AFOSR grant FA8655-13-1-3084, and NSERC CGS-M. We would like to thank the Centre for Theoretical Neuroscience of the University of Waterloo for providing travel grants, Ménélik Vero for gathering the training and test data, Alexander Pekarovskiy and Michaela Semmler for valuable comments, Mitsubishi Electric Corporation for donating the RV-2F-D robotic arm, and Rintaro Haraguchi for his assistance in programming the C++ interface.

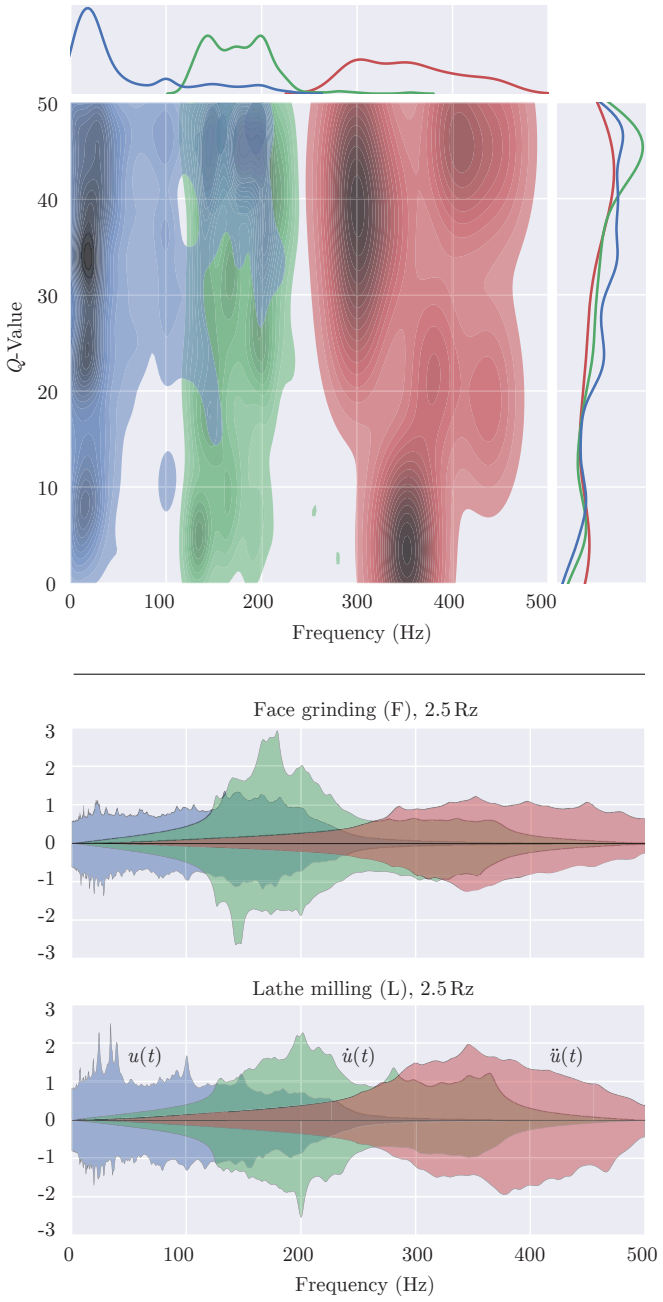

Fig. 5. Visualizing the network in the frequency domain. Frequencies are $0-250$, $125-375$, or $250-500$ Hz, depending on whether each hidden neuron is encoding a dimension from $u(t)$ (blue), $\dot{u}(t)$ (green), or $\ddot{u}(t)$ (red). (Top) Distribution of bandpass filter parameters from (6), weighted by the ℓ^2-norm of SVM coefficients, and smoothed by a kernel density estimate [26]. The smallest weights are omitted to reduce visual clutter. Histograms along the sides flatten the distribution across their corresponding axes. (Bottom) Power of bandpass filters per texture (only two shown), weighted by their squared SVM coefficients (unitless). Negatively rectified dimensions are flipped about the x-axis for visualization.

References

[1] B. Unger, R. Klatzky, and R. Hollis, "The physical basis of perceived roughness in virtual sinusoidal textures," *Haptics, IEEE Transactions on*, vol. 6, no. 4, pp. 496–505, 2013.

[2] R. L. Klatzky, D. Pawluk, and A. Peer, "Haptic perception of material properties and implications for applications," *Proceedings of the IEEE*, vol. 101, no. 9, pp. 2081–2092, 2013.

[3] M. Holliins, R. Faldowski, S. Rao, and F. Young, "Perceptual dimensions of tactile surface texture: A multidimensional scaling analysis," *Perception & psychophysics*, vol. 54, no. 6, pp. 697–705, 1993.

[4] A. I. Weber, H. P. Saal, J. D. Lieber, J.-W. Cheng, L. R. Manfredi, J. F. Dammann, and S. J. Bensmaia, "Spatial and temporal codes mediate the tactile perception of natural textures," *Proceedings of the National Academy of Sciences*, vol. 110, no. 42, pp. 17 107–17 112, 2013.

[5] R. S. Dahiya, G. Metta, M. Valle, and G. Sandini, "Tactile sensing-from humans to humanoids," *Robotics, IEEE Transactions on*, vol. 26, no. 1, pp. 1–20, 2010.

[6] R. D. Howe and M. R. Cutkosky, "Dynamic tactile sensing: Perception of fine surface features with stress rate sensing," *Robotics and Automation, IEEE Transactions on*, vol. 9, no. 2, pp. 140–151, 1993.

[7] P. Giguere and G. Dudek, "A simple tactile probe for surface identification by mobile robots," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 534–544, 2011.

[8] J. Sinapov, V. Sukhoy, R. Sahai, and A. Stoytchev, "Vibrotactile recognition and categorization of surfaces by a humanoid robot," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 488–497, 2011.

[9] Y. Tada, K. Hosoda, and M. Asada, "Sensing ability of anthropomorphic fingertip with multi-modal sensors," in *IEEE International Conference on Intelligent Robots and Systems*. Citeseer, 2004, pp. 1005–1012.

[10] N. Jamali and C. Sammut, "Majority voting: material classification by tactile sensing using surface texture," *Robotics, IEEE Transactions on*, vol. 27, no. 3, pp. 508–521, 2011.

[11] L. Bologna, J. Pinoteau, J. Passot, J. Garrido, J. Vogel, E. R. Vidal, and A. Arleo, "A closed-loop neurobotic system for fine touch sensing," *Journal of neural engineering*, vol. 10, no. 4, p. 046019, 2013.

[12] U. B. Rongala, A. Mazzoni, and C. M. Oddo, "Neuromorphic artificial touch for categorization of naturalistic textures." *IEEE transactions on neural networks and learning systems*, 2015.

[13] J. A. Fishel and G. E. Loeb, "Bayesian exploration for intelligent identification of textures," *Frontiers in neurorobotics*, vol. 6, 2012.

[14] S. S. Kim, S. Mihalas, A. Russell, Y. Dong, and S. J. Bensmaia, "Does afferent heterogeneity matter in conveying tactile feedback through peripheral nerve stimulation?" *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 19, no. 5, pp. 514–520, 2011.

[15] H. P. Saal and S. J. Bensmaia, "Touch is a team effort: interplay of submodalities in cutaneous sensibility," *Trends in neurosciences*, vol. 37, no. 12, pp. 689–697, 2014.

[16] H. Jörntell, F. Bengtsson, P. Geborek, A. Spanne, A. V. Terekhov, and V. Hayward, "Segregation of tactile input features in neurons of the cuneate nucleus," *Neuron*, vol. 83, no. 6, pp. 1444–1452, 2014.

[17] C. Eliasmith and C. H. Anderson, *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press, 2003.

[18] A. Mundy, J. Knight, T. C. Stewart, and S. Furber, "An efficient spinnaker implementation of the neural engineering framework," *IJCNN*, 2015.

[19] J. Hasler and B. Marr, "Finding a roadmap to achieve large neuromorphic hardware systems," *Frontiers in neuroscience*, vol. 7, 2013.

[20] T. Bekolay, J. Bergstra, E. Hunsberger, T. DeWolf, T. C. Stewart, D. Rasmussen, X. Choo, A. R. Voelker, and C. Eliasmith, "Nengo: a python tool for building large-scale functional brain models," *Frontiers in neuroinformatics*, vol. 7, 2013.

[21] E. Hunsberger, M. Scott, and C. Eliasmith, "The competing benefits of noise and heterogeneity in neural coding," *Neural Computation*, vol. 26, no. 8, 2014.

[22] C. Koch, *Biophysics of computation: information processing in single neurons*. Oxford university press, 1999.

[23] R. S. Johansson and J. R. Flanagan, "Coding and use of tactile signals from the fingertips in object manipulation tasks," *Nature Reviews Neuroscience*, vol. 10, no. 5, pp. 345–359, 2009.

[24] R. Jolivet, R. Kobayashi, A. Rauch, R. Naud, S. Shinomoto, and W. Gerstner, "A benchmark test for a quantitative assessment of simple neuron models," *Journal of neuroscience methods*, vol. 169, no. 2, pp. 417–424, 2008.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] M. Waskom, O. Botvinnik, P. Hobson, J. Warmenhoven, J. B. Cole, Y. Halchenko, J. Vanderplas, S. Hoyer, S. Villalba, E. Quintero, and et al., "seaborn: v0.6.0 (june 2015)," 2015. [Online]. Available: http://dx.doi.org/10.5281/zenodo.19108