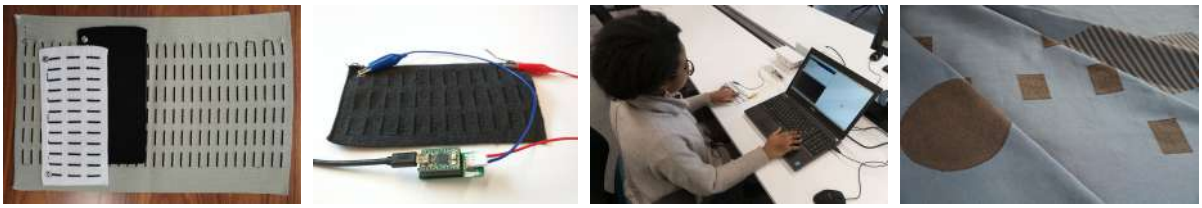
Knitted Sensors: Designs and Novel Approaches for Real-Time, Real-World Sensing

DENISA QORI MCDONALD, College of Computing and Informatics, Drexel University, USA
RICHARD VALLETT, Center for Functional Fabrics, Drexel University, USA
ERIN SOLOVEY, Worcester Polytechnic Institute, USA
GENEVIÈVE DION, Center for Functional Fabrics, Drexel University, USA
ALI SHOKOUFANDEH, College of Computing and Informatics, Drexel University, USA



(a) (b) (c) (d)

Fig. 1. Knitted sensor designs and applications. *(a)*: Rectangular touchpads of varying size, knitted with carbon-coated nylon and polyester yarn. *(b)*: The real-time sensing controller connected to a knitted touchpad through electrodes at the yarn endpoints. *(c)*: Data collection performed with a 36-button knitted touchpad. *(d)*: Three knitted fabric prototypes of varying patterns composed of the same carbon-coated nylon yarn, and polyester yarn.

Recent work has shown the feasibility of producing knitted capacitive touch sensors through digital fabrication with little human intervention in the textile production process. Such sensors can be designed and manufactured at scale and require only two connection points, regardless of the touch sensor form factor and size of the fabric, opening many possibilities for new designs and applications in textile sensors. To bring this technology closer to real-world use, we go beyond previous work on coarse touch discrimination to enable fine, accurate touch localization on a knitted sensor, using a recognition model able to capture the temporal behavior of the sensor. Moreover, signal acquisition and processing are performed in real-time, using swept frequency Bode analysis to quantify distortion from induced capacitance. After training our network model, we conducted a study with new users, and achieved a *subject-independent accuracy of 66%* in identifying the touch location on the 36-button sensor, while chance accuracy is approximately 3%. Additionally, we conducted a study demonstrating the viability of taking this solution closer to real-world scenarios by testing the sensor's resistance to potential deformation from everyday

Authors' addresses: Denisa Qori McDonald, dq38@drexel.edu, College of Computing and Informatics, Drexel University, 3675 Market St. Philadelphia, Pennsylvania, USA; Richard Vallett, rjvallett@drexel.edu, Center for Functional Fabrics, Drexel University, 3101 Market St. Philadelphia, Pennsylvania, USA; Erin Solovey, esolovey@wpi.edu, Worcester Polytechnic Institute, 100 Institute Rd, Worcester, Massachustes, USA; Geneviève Dion, Center for Functional Fabrics, Drexel University, 3101 Market St. Philadelphia, Pennsylvania, USA; Ali Shokoufandeh, College of Computing and Informatics, Drexel University, 3675 Market St. Philadelphia, Pennsylvania, USA.

2474-9567/2020/12-ART145 $15.00
https://doi.org/10.1145/3432201

conditions. We also introduce several other knitted designs and related application prototypes to explore potential uses of the technology.

CCS Concepts: • **Computing methodologies → Neural networks**; *Machine learning algorithms*; *Supervised learning by classification*; Cross-validation; • **Human-centered computing → Interaction techniques**; **Ubiquitous and mobile computing**; *Interaction devices*; **User studies**; *Usability testing*.

Additional Key Words and Phrases: knitted sensors, neural networks, real-world sensing, real-time sensing, touch detection, evaluation study, robustness study

ACM Reference Format:
Denisa Qori McDonald, Richard Vallett, Erin Solovey, Geneviève Dion, and Ali Shokoufandeh. 2020. Knitted Sensors: Designs and Novel Approaches for Real-Time, Real-World Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 145 (December 2020), 25 pages. https://doi.org/10.1145/3432201

1 INTRODUCTION

The ubiquity of textiles, as well as their soft and flexible nature, makes them an appealing medium for tangible embedded interactions. Textile touch sensors have been created through various manufacturing techniques such as weaving [29], embroidery [28], and knitting [32] to investigate their efficacy toward creating flexible interfaces. The interaction potential of textile sensors has been demonstrated through the introduction of various application prototypes and use cases [14, 24, 29, 39]. However, for textile-based touch sensors to conform to users' expectations and have real-world viability, there are many practical considerations that must be addressed. One important aspect is designing toward scalable manufacturing to create textile devices that take advantage of mass-production techniques and that use commercially-available materials. Another is recognizing that the physical demands regularly placed on textiles must not interfere with the device's performance. In particular, textile touch sensors must maintain integrity and sensitivity, while easily flexing, folding, stretching, and dynamically conforming to the human body in motion [9]. These needs, however, are not satisfied by many existing fabric touch sensors that model their circuit designs on conventional touch interfaces. Such sensors rely on a multitude of wires or relatively complex, connected layers to sense input. This intricate construction reduces durability and leaves such sensors susceptible to damage and deterioration from bending, especially at fabric-to-wire connection points.

To address durability and connectorization, as well as production at scale, we leverage digital weft knitting; a manufacturing solution that produces textiles using an array of continuous, intermeshed yarns. Weft knitting enables the fabrication of a textile circuit with a single, electrically continuous trace. Carbon-suffused nylon yarn is used as the conductor; it is combined with non-conductive yarns throughout the fabric structure. The fabric circuit design is directly compatible with the weft knitting process and does not require any post-assembly. There are only two electrical connections per sensor—one at each end of the knitted electrical pathway—which simplifies the connection to sensing hardware and affords greater textile flexibility and connection durability. Furthermore, the circuit pattern can be modified or scaled to produce form factors of different shapes and sizes, while requiring no modifications to sensing hardware.

These practical benefits come at the expense of easily and precisely localizing touch. Unlike discrete-electrode circuits that use many individual wires to localize distributed touch, this circuit relies on only two signals measured at each endpoint of the conductive yarn. Therefore, more complex signal processing techniques and computational models are needed to identify location of touch. Moreover, in order for a user interface to be natural and usable, the requirement of user-specific calibration should be avoided, especially in cases when the same sensor is expected to be used by multiple people. Early work showed an initial proof-of-concept system that coarse locations could be differentiated on a sensor pad, but highlighted the challenge of precise touch

localization [37]. Further work [36] focused on high-fidelity and invariant signal representation with respect to touch location, but location identification was not performed, and signal processing was performed offline.

In this work, we advance the technical development of a weft-knitted, fabric-based capacitive touch sensor. We present a real-time sensing system and a classification model that accurately localizes touch along the yarn in real-time, under real-world conditions. We also introduce several other sensor designs, which aim to be inconspicuous and minimalistic, intuitive to users, as well as valuable for interactive experiences, using digital weft knitting. This work brings us closer to real-world, real-time, and scalable textile sensing by making the following contributions:

- A *recognition model* to identify precise location of touch on an existing knitted sensor circuit relying on one conductive yarn, without subject-specific calibration. This model uses a statistical, frequency-based feature construction component and an LSTM neural network classifier, addressing the challenges of minimal information output from the single-yarn system, as well as noise in the signal.
- A *real-time sensing and signal processing circuit* which obtains input from the knitted sensor. It uses Bode analysis to reject the frequencies most prone to electromagnetic interference (EMI).
- Results from a *series of user studies* to move toward real-time, real-world knitted sensors. The studies enable the creation of a robust location sensing model; demonstrate its generalizability with new users; and investigate its robustness under real-world conditions.
- *Design and application prototypes* of knitted sensors illustrating the versatility and scalability of the knitting design process using a single conductive yarn, and the usability of knitted sensors in diverse interactive applications. Some of these design form factors have the potential to be interchangeably used with our sensing system and recognition infrastructure, provided each one relies on its own individually-trained location identification model.

2 BACKGROUND AND RELATED WORK

The development of fabric-based flexible touch sensors spans over two decades. Initial research examined the feasibility of touch-sensitive fabrics as a novel method of human-computer interaction [28]. Current research investigates the use of fabrics as robust and extensible interfaces, capable of augmenting existing materials or devices [29]. Creating an extensible textile touch sensing system requires reconciling factors beyond fabric construction. Factors such as machine learning, signal processing, and novel sensing methods must be considered to create platforms that provide a foundation for continued development.

This section describes prior research aiming to create textile touch sensing systems. In Section 2.1, we summarize concepts related to different ways of creating smart fabrics, focusing on the intersection between fabric construction techniques and their associated sensing methodologies. Background on knitting and large-scale manufacturing in Section 2.2 helps elucidate some of our choices and challenges in creating knitted sensors and associated systems. Next, in Section 2.3, we discuss common sources of distortion that affect electrical circuits, and their effects on accurate touch localization in systems such as ours. In Section 2.4, we provide some background on the neural network structures on which we rely to create our model.

2.1 Textile Touch Sensing and Assembly Approaches

2.1.1 Textile Touch Sensing Approaches. Textile sensors detect touch using conductive yarns by sensing changes in inter-yarn *contact*, *resistance*, or *capacitance*.

Contact sensing operates by detecting the presence or absence of contact between two conductive yarns. This touch detection method has been used to create large-scale distributed fabric-based touch sensors [19]. Physical contact deforms a layered fabric structure to make or break inter-yarn contact between conductive layers to sense normal touch. Minimalistic applications of contact sensing have also been used to create novel, *single-layer*

fabric-based user interfaces. *Pinstripe* [21], a textile user interface element on smart garments, uses pinching and rolling of cloth between the user's fingers to measure input. The sensing detects changes relative to the displacement of the two sides of the fold, measured by conductive threads sewn into the fabric, and interpreted as a continuous change in value.

Resistive sensing measures touch through electrical resistance changes caused by yarn strain or deformation. Sundholm et al. [34] introduce *Smart-mat*, a resistive pressure-sensing matrix incorporated into an exercise mat to recognize physical exercises performed on it. This sensor is composed of two layers of striped conductive foil with a conductive polymer fiber sheet in between, forming a 80×80 electrode grid. *Piezoresistive yarn* is used in sensors that measure contact as a change in resistance, and has been explored in conjunction with several textile assembly techniques, such as hand sewing, machine sewing, weaving, and knitting [24, 26]. Conductive polymerization using piezoresistive materials has been explored to realize interactive e-Textiles [16]. The outcome of the polymerization process varies largely depending on fabric type, making the optimal design more of a trial and error process, but highly repeatable once the optimal conditions are identified. *Second Skin* [8], a flexible and stretchable electronic textile garment that fits the body, uses a multi-layer construction approach and heat-reactive precision bonding films for fusion of materials. This design is creative, functional, and shows potential for personal and industrial manufacturing, once there is further progress on implementation of the manufacturing process and sensing strategy.

Capacitive sensing measures the charge stored between two parallel conductors, such as two wires, or a wire and human skin. Capacitive sensors have an advantage of high sensitivity, low cost, and low profile compared to both *contact* and *resistive* sensors. A basic textile capacitive sensor that measures human skin contact through *self-capacitance* is among the simplest circuits to produce in fabric—requiring only one conductive yarn and external connection to measure a discrete touch location. Our work utilizes *capacitive* sensing to detect human skin contact. Since our sensors rely on one continuous conductive yarn, we need to consider methods that localize touch on a single wire, such as time-domain reflectometry (TDR), frequency-domain reflectometry (FDR), and differential capacitive sensing.

Time- and frequency-domain reflectometry (TDR and FDR) are methods that localize touch by measuring the distance-to-fault, such as a change in capacitance, across a transmission line (Figure 2(c)). Promising applications of this technology are shown in [17, 18, 39]. However, reflectometry requires a high sampling rate to yield accurate touch location, and the complexity of the wire structure together with the computationally-intensive sensing method, can pose practical challenges when using in textile applications.

Differential capacitive sensing measures both the magnitude and location of induced capacitance using a current differential along a linear conductive pathway [36, 37] (Figure 2(d)). Differential capacitive sensing also captures applied pressure during touch, an aspect that can be potentially useful for a variety of applications.

2.1.2 Textile Assembly Approaches and Touch Sensing. Textile assembly or embellishing methods, such as embroidery, weaving, and knitting have been used to create touch-sensitive textile devices using conductive yarn and we describe characteristics related to touch sensing of each below.

Embroidery is a textile embellishing method that inserts thread onto a fabric substrate to create arbitrary patterns (Figure 2(a)). It was among the first methods used to create a textile touch sensor [28], and has since been used to create interactive textile devices [11, 13, 14]. Since embroidered threads are independent of the original textile, they can overlap and be routed in arbitrary directions to create a variety of patterns. Embroidery as an embellishment technique, however, is not an integral part of fabric construction.

Large-area distributed touch sensing can also be achieved through integrating conductive yarns into the textile structure during formation, eliminating post processing. *Weaving* and *knitting* are textile manufacturing processes that can combine conductive and non-conductive yarns to form fabric circuits. *Knitting* produces fabric by employing a continuous yarn or set of yarns to form a series of interlocking loops. Knitting has been recently

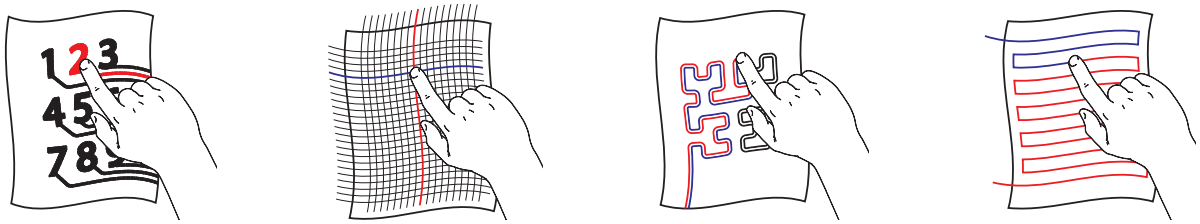


Fig. 2. Illustrations of sensing techniques and electrode layouts used in textile touch sensors. *(a)*: Embroidered capacitive buttons. *(b)*: Woven or embroidered capacitive sensing matrix. *(c)*: *Reflectometry* measured using two parallel conductors. *(d)*: Single weft knitted conductor measured using *differential capacitive sensing*.

explored in producing interactive textiles, such as force-sensing knitted structures [27], knitted electronic musical controllers [38], and knitted dynamic displays [7]. Other research has investigated creating 2D sensing structures using a homogeneous [36, 37] or multi-material [23] 1D filament.

Weaving is the process of interlacing perpendicular yarns to form a textile (Figure 2(b)). It is a widely-used method to produce distributed textile touch sensors [2, 6, 15, 29, 35] due to its similarity to standard capacitive and resistive sensing *matrices*. Woven textiles use the perpendicular alignment of numerous *weft*—horizontal and *warp*—vertical yarns to form a sensing grid and detect touch at yarn intersection points. Because yarns can be tightly packed within a weave, a woven touch sensor can support a high-resolution distributed sensing area. However, connecting a matrix to external sensing electronics requires an equal number of fabric-to-wire connections—a difficult and time-consuming post production process. These connections often exist at millimeter-scales and utilize techniques like soldering, which create brittle joints that reduce the textile's durability. This necessity can be impractical for longevity and for real-world use. One potential solution towards creating durable fabric-based touch sensors is to decrease the number of required external fabric-to-wire connections. Using fewer connections improves the flexibility of the textile and reduces the likelihood of failure of any individual connection.

Single-wire distributed sensing methods, including *TDR, FDR* and *differential capacitive sensing*, have been explored as solutions to limit the number of required fabric-to-wire connections—using as few as two connections to external sensing hardware. *Differential capacitive sensing* has shown promise to coarsely localize touch along a single conductive yarn, while retaining the benefits of single-wire approaches and enabling simple wire structures. This method is especially suited for *weft-knitting*, which can create fabrics from single, continuous yarns. *Differential capacitive sensing* is compatible with these sensors, since it localizes touch on a single, linear conductor. Other work [36] related to single-yarn knitted sensors and *differential capacitive sensing* has presented algorithmic advances related to invariant signal representation, but not classification of location of touch. Moreover, the signal processing work related to this technique, so far has been done offline [36, 37]. Our work aims to extend that research to enable real-world use, when integrated with streamlined manufacturing processes.

2.2 Streamlined Manufacturing of Knitted Sensors: Benefits and Design Requirements

Aligning touch-sensitive textile construction with industry manufacturing standards enables reproduction of textiles on machinery outside of a laboratory, an aspect addressed by our design and others [4, 29, 36–38]. In order to improve viability for large-scale production of our textile touch sensors, we use digital weft-knitting, an industry-standard, automated, end-to-end textile manufacturing process that can produce complete forms with minimal human intervention [36, 37]. It is beneficial to smart fabric construction for the following reasons:

(1) Knitting integrates both conductive and non-conductive yarns within the textile structure to create the touch sensing circuit.
(2) Digital knitting machines can produce complete textiles without the need for manual operations or human intervention during or after assembly, including positioning of sensing elements within the textile.
(3) The loop structure of digitally knitted textiles is controlled programmatically. We can specify a variety of knitted forms with minimal reprogramming and machine reconfiguration.
(4) Digital weft-knitting machines can reproduce identical knitted textiles, including electrical circuits with similar properties.

The digital weft-knitting design process addresses shortcomings in the construction of many fabric-based sensors, namely in reducing the number of fabric-to-wire connections between the textile and sensing hardware, and in supporting large-scale production, while permitting design modifications. However, this manufacturing process has several physical requirements regarding yarn placement during textile construction that must be considered:

(1) The yarn must remain contiguous within the textile, and must not separate or break within the textile. However, it may be cut at the edge, or after it has exited, and before it re-enters the fabric.
(2) Any yarn entering the textile must exit the textile.
(3) Yarn deposition can only take place horizontally and upward. Horizontal rows of loops are sequentially assembled from the bottom of a textile to the top. Loops are dropped from holding hooks when a new loop is formed and are irretrievable by the knitting machine.

These requirements inform the design of the touch-sensitive circuits using this method.

2.3 Addressing Distortion in Single-wire Touch Localization

We rely on *differential capacitive sensing* as used jointly with knitting to produce smart fabric. Capacitive sensing is susceptible to distortion due to the low current used to detect changes in capacitance and the proportionally large response caused by fluctuations in voltage. Additive interference originates from both device-centric sources like *power supply ripple* and external sources like *electromagnetic interference*. In estimating location across electrically-continuous substrates using differential capacitive sensing, additive interference reduces the detection rate of touch, and therefore the touch localization accuracy. In this way, otherwise insignificant sources of uncertainty, produce considerable measurement variance that inhibits practical use.

Bode analysis [3] used with differential capacitive sensing has been introduced as a potential solution for removing recorded interference [36]. Bode analysis is an extension of Fourier analysis, which transforms a measured time-domain signal into frequency-domain components. Bode analysis compares both the ratio of a system's Fourier transform complex magnitude output to its input, and the phase difference between output and input. The frequencies present in an input will be present in an output. Given an input with a known frequency composition, it is possible to reconstruct the output while bypassing additive frequencies.

2.4 Recurrent Neural Networks

The time-series data acquired through the two connections of the sensor needs to be processed to map to a location of touch. The classification algorithm should take advantage of the continuity of sampled frequencies over time, in order to be able to properly model the signal from the two connections. Moreover, due to variability in the data, such as different users, conditions, finger placement and more, key presses of the same location should be represented in terms of essential patterns within them. To those ends, we utilize Recurrent Neural Networks (RNNs) [31], which have found application for many classification tasks. We see promise in using them in a neural network architecture to identify location of touch, given the signal data from each electrode connected to our knitted sensors as an input.

RNNs allow previous outputs to be utilized as inputs in addition to hidden states of traditional multi-layer perceptrons (MLP). This flexibility makes them ideal for classification of input of any length, while maintaining limits on model size, taking into account historical information, and sharing weights across time. A known drawback of RNN models is their difficulty in capturing long term dependencies due to multiplicative gradient. The error of such models can be exponentially decreasing/increasing with respect to the number of layers. The Gated Recurrent Unit (GRU) [5] and their generalizations, such as Long Short-Term Memory units (LSTM) [10], address to some extent the vanishing gradient problem encountered by traditional RNNs. We use LSTMs to model the temporal dependencies in our data.

3 TOWARD ACCURATE REAL-TIME, REAL-WORLD SENSING

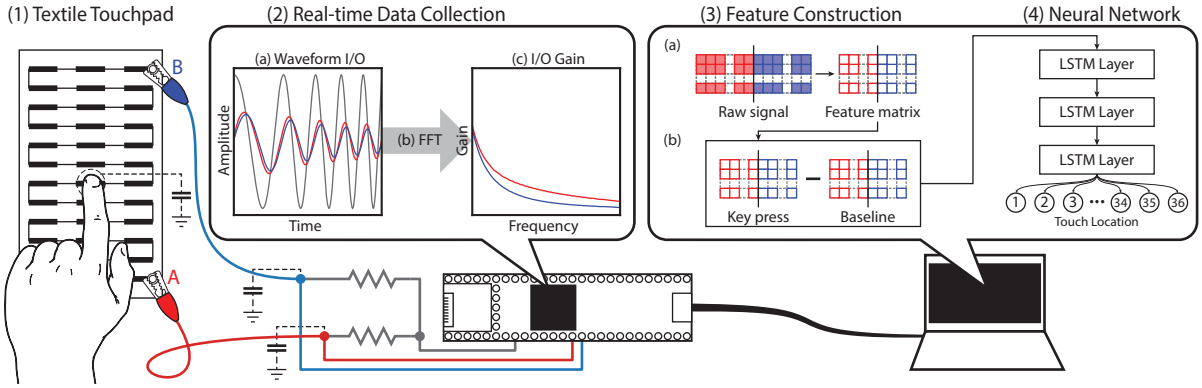


Fig. 3. Diagram of the interaction pipeline of the system components: (1) The knitted touchpad is pressed at one of the pre-defined touch locations; (2) voltage is sourced and measured at endpoints A and B and transformed within the embedded sensing microprocessor to yield the output-to-input gain ratio per frequency; (3) features are constructed based on statistics of the frequency response of measured touch data; (4) the feature representation is used as an input to a neural network to predict the discretized touch location.

In this work, we focus on building a model to enable precise, real-time touch localization on an existing knitted sensor which relies on one conductive yarn as a sensing element. To bring such sensors into real-world use, in addition to the sensing fabric, we design our system architecture with two main components: the *recognition model* to determine the position of touch on the knitted sensor, which would be able to receive data from our *real-time signal acquisition and processing system*. The model, once trained for high-accuracy recognition, can, in the future, be embedded into a dedicated microprocessor or handheld device, and connect to the signal acquisition and processing controller to identify location of touch in real time. Figure 3 provides an overview of the components involved in this process.

For our real-time system and recognition model, we use a knitted pattern previously introduced [36, 37], on which it is relatively difficult to accurately resolve location of touch, compared to other possible designs. The minimalistic design of the fabric circuit, which simplifies manufacturing and enhances usability, requires complex computational methods to achieve location accuracy. The main challenge in classifying touch location for this sensor's design pattern is that all locations across a 2-D space are defined along a continuous 1-D yarn pathway. Moreover, only voltage readings at each endpoint of the yarn are used as information from the system. The signal data obtained is also subject to considerable noise. Common sources of additive electrical distortion, such as power supply ripple and electromagnetic interference, introduce significant variance in measured voltage.

This variance reduces effective touch location identification accuracy. Its effects can be mitigated through touch sensor design by producing circuits with large inter-touch-point resistances, as we discuss in Section 5 below. However, it is necessary to address the effects of distortion through signal processing and machine learning to ease the constraints placed on physical textile design and allow for a larger variety of design forms. In addition to these aspects that make high-accuracy location identification a challenging task, we would want to use the sensor without user-specific calibration, for a more intuitive and user-friendly interaction.

Previous research [36] demonstrated the technical feasibility of constructing an unsupervised system that can differentiate between touch locations on the knitted sensors with one sensing yarn. It proposed a model for extracting invariant features, *Mixed-Source Description (MSD)*, corresponding to key-press events, and showed that key-press events as represented by MSD features of the same key, are more similar to each other than key-press events of different keys. To compute the similarity of MSD features, the *Euclidean Levenshtein Distance (ELD)* measure was introduced, which can compare tensors of varying lengths. However, these results did not provide a location of touch. Instead, that work focused primarily on the analysis of the system and of the proposed algorithm, demonstrating the ability to represent the signal in a way that has the potential to later lead into extracting location of touch. Similarly to our work, Bode analysis was utilize to quantify the attenuation of input voltage waveforms at specific frequencies. The raw signal was the system's response to one frequency input through Bode analysis. However, other frequencies would also be helpful in capturing changes differently, collectively offering a better picture of the system behavior than a single frequency. Moreover, previous signal processing was performed offline for that task.

We extend the work in [36] by constructing a real-time signal processing system, using a swept-frequency excitation signal to perform Bode analysis, instead of a single-frequency signal, and adding a learned model to identify touch locations. We first construct features over the signal, and then train our model using a multi-layer Long Short Term Memory (LSTM) network architecture, designed to capture the temporal dependencies within the input signals, and classify touch location. This section describes the components of the real-time signal processing system, and those of the recognition model for location identification.

3.1 Real-Time Signal Acquisition and Processing System

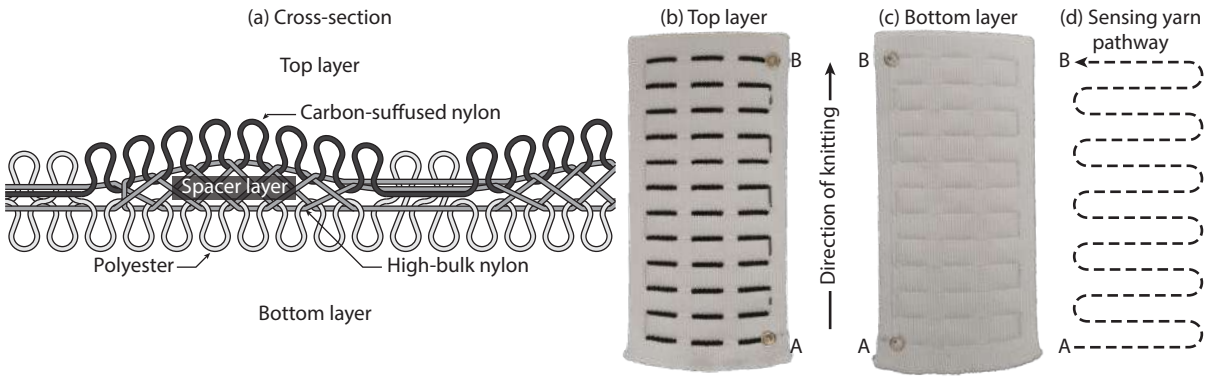


Fig. 4. Weft-knitted touch sensor design and construction. *(a)*: Illustration of the layering formed by digital flatbed weft knitting with conductive carbon-coated nylon yarn, and non-conductive polyester and high-bulk nylon yarns. *(b)*: Image of the top layer of a completed sensor. A and B denote the yarn endpoints, where external electrodes can be connected. Darker lines indicate touch areas. *(c)*: Image of the bottom layer of a completed sensor. There are no sensing elements on this layer. *(d)*: Illustration of the serpentine pathway of the carbon-suffused nylon formed by the knitting process.

3.1.1 Knitted Circuit. Figure 4a illustrates the cross-section showing the yarn layers formed during the weft knitting process. The conductive and non-conductive yarns form distinct but interconnected layers that compose both the textile structure and the overall linear circuit. Our recognition model is trained to classify the location of touch specifically for the rectangular touch sensor depicted in Figures 4b and 4c. The sensor geometry defines 12 rows and 3 columns of exposed conductive yarn pads, representing buttons, for a total of 36 distinct touch locations. The locations are evenly distributed across the textile surface in the serpentine pathway shown in figure 4d and form a series-connected circuit having uniformly distributed inter-touchpoint resistance. The linear distance between the center of each touch location is approximately 1 inch. We aim to distinguish between touching each of them. For initial evaluation, we limit contact to single-touch during data collection.

This rectangular button layout is useful for generic touch applications, like measuring taps or swipes. However, series-connected adjacent buttons with a low inter-button resistance pose challenges towards distinction, which reduces accurate touch location identification [36, 37]. Touch locations are distinguished by a current differential caused by a difference in resistance. This touch sensor's cumulative linear resistance is approximately 550 kΩ with the center-point of each touch location separated by an average resistance of 15 kΩ.

Other sensor forms could be easily swapped with the existing one, however, the neural network would need to be trained for each sensor form individually, since the interactive components (the exposed carbon-suffused nylon touch areas) of the sensor would change depending on the design and needs of the application.

3.1.2 Embedded Sensing Microprocessor. The sensing hardware uses an NXP Kinetis® MK66 ARM® Cortex®-M4F 180 MHz microprocessor (PJRC Teensy 3.6 development board), shown in the second part of Figure 3. Two connections lead the fabric circuit to the sensing hardware, which can connect to a computer via USB and transfer processed data at approximately 4.3 Mbaud, and 125 Hz to a *Processing* [30] application for data visualization.

Bode analysis [3] is a system identification approach, used to characterize the behavior of an unknown system given controlled inputs. It measures the amplitude attenuation and phase shift between the Fourier transforms of the system's input and output as a function of frequency. Our work goes beyond previous work that used only the attenuation of a single frequency [36] and instead considers the contributions of attenuation across multiple frequencies. This provides a greater amount of information to describe touch, and also accounts for variance in any single frequency bin through consensus. In order to control the frequency composition of the input waveform, a linear swept-frequency cosine wave is generated according to Equation (1). This swept-frequency signal has a start and end frequency of f_0 and f_1 Hz respectively and has a time duration between t_0 to t_1 seconds. The generated waveform is stored in memory and outputted by the microprocessor's internal 12-bit, 0 V to 3.3 V digital-to-analog converter (DAC):

$$c(t) = b + A\cos\left(2\pi f_0 t + \frac{(f_1 - f_0)\pi t^2}{t_1}\right) \tag{1}$$

Figure 5 illustrates data flow within the microprocessor. Two 16-bit analog-to-digital converters (ADCs) synchronized with the DAC sample the output waveform at approximately 256k samples-per-second in 1024-point increments. Fast-Fourier Transforms of the samples were processed using the ARM CMSIS DSP library complex FFT function [22]. Ninety six frequencies between $f_0 = 2,000$ Hz and $f_1 = 26,000$ Hz were input for a duration of $t_0 = 0$ seconds to $t_1 = 0.004$ seconds. The waveform uses the full range of the ADC's voltage output, between 0 V and 3.3 VDC, thus the amplitude, A, and bias, b, are 1.65 V and 1.65 V respectively. To characterize the system response during data analysis, we use the gain values of all recorded frequencies, for a total of 192 values per record window.

The swept-frequency range is chosen based on the locations of frequencies affected by additive distortion (Figure 5e). The two significant sources of distortion are *power supply ripple*, contributed by the microprocessor's DC power source, and *electromagnetic interference* (EMI), induced by external power sources in proximity. Power

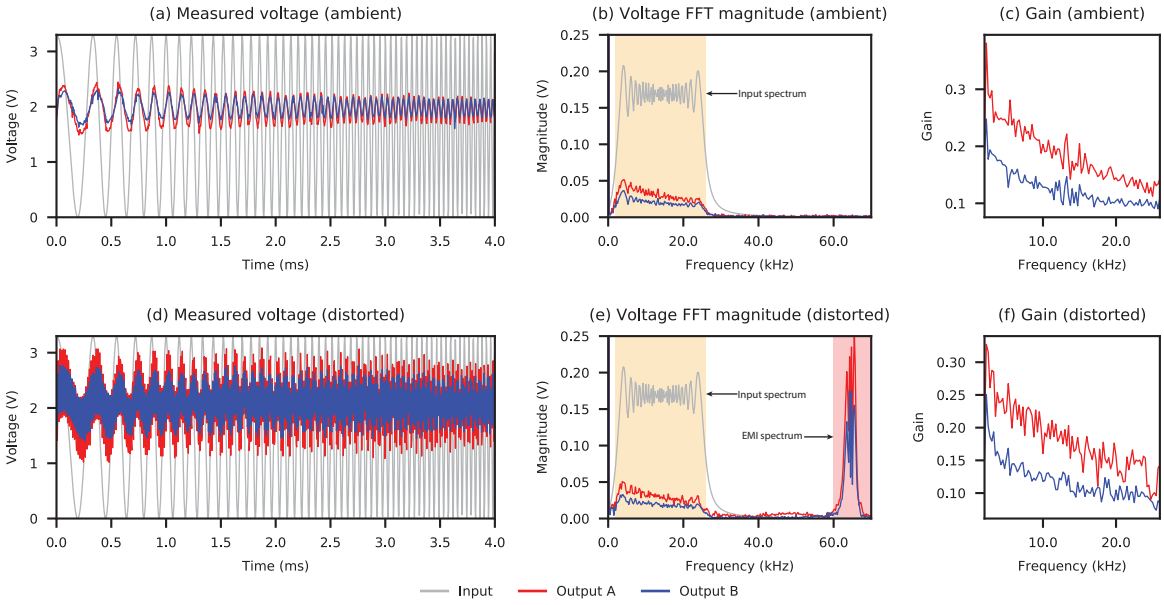


Fig. 5. Data processing within the microprocessor in ambient conditions, illustrated in sub-figures *a*, *b*, and *c*, and under exposure to EMI, as seen from sub-figures *d*, *e*, *f*. *(a)* A swept-frequency cosine wave is passed through current-limiting resistors and input at both endpoints of the sensing pathway. The voltage outputs (red, blue) are measured at the endpoints. *(b)* The magnitude of all three waveforms is computed via FFT. *(c)* The output gain is computed as a function of frequency. *(d)* The same process described in *(a)*, but under exposure to EMI. *(e)* The same process described in *(b)*, but under exposure to EMI. *(f)* The same process described in *(c)*, but under exposure to EMI.

supplies that transform AC to DC voltage often induce a small amount of ripple into the rectified output. Ripple is a small fluctuation in the device's operating voltage, measured at approximately 200 mV at 60 Hz in ambient conditions.

Human touch contributes to measured distortion, increasing the ripple amplitude to approximately 1 V. Because capacitive sensors rely on a voltage difference to charge and discharge a capacitor in contact with the sensor, and induce a controlled oscillation, this phenomenon contributes to the induced oscillation to create unexpected voltage shifts in the time domain. Electromagnetic interference is radiated distortion produced by circuits transmitting power. The severity to which the EMI affects surrounding electronic devices depends on the quality of EMI shielding. Capacitive sensing relies on exposed, unshielded wiring to measure touch, and thus is more susceptible to EMI [33]. The distortion shown in Figures 5d and 5e is generated by a 70 Watt fluorescent lamp in close proximity, approximately 18 in., to the fabric sensor. An interference band of 4 kHz centered at 40 kHz was observed as a resultant of the EMI. The proximity of the lamp affects the induced distortion.

3.2 Recognition Model

The purpose of the recognition model is to translate the acquired signal data into one of the 36 discrete interactive components of the knitted touchpad. Our approach is composed of two main components: a feature construction step, and the neural network for classification.

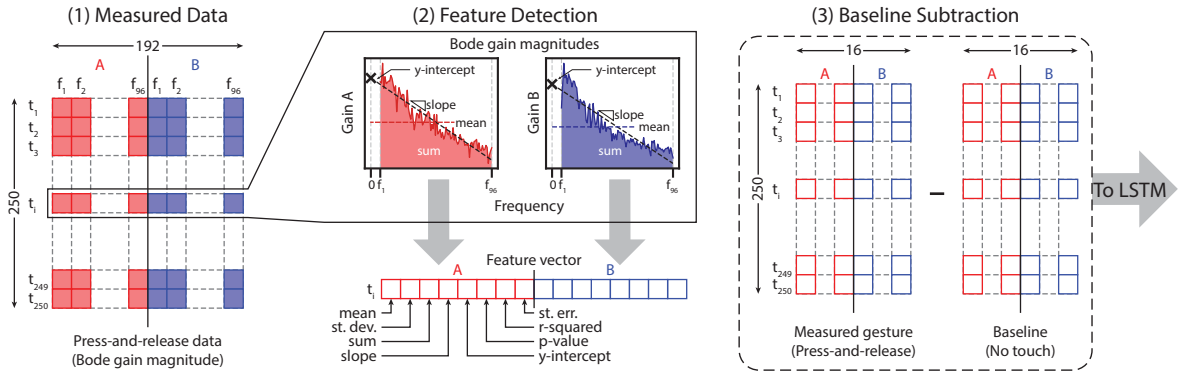


Fig. 6. Feature construction over frequency gain values of both signals: (1) the raw data is composed of 250 time instances and 96 frequency values per signal, for a total of 192; (2) statistical features are computed over the frequencies per time step; (3) baseline is subtracted from key-press data.

3.2.1 Feature construction. The system constructs features from the input-to-output voltage gain values of the sensor over time. A touch event is measured over two seconds. This allows enough time to capture the baseline, onset, saturation, and offset. Each touch event consists of 250 temporally-related measurements of the voltage gain values, which are captured as explained in Section 3.1.2. This amount of time is more than enough for the voltage discharge which defines the touch event to occur. Since there are 96 frequency responses recorded from each of the two sensor endpoint signals, one touch sample is represented as a 250x192 matrix before feature construction, associated with its respective touch location or *button* number as a label. We construct statistical features over all the captured frequency responses for each signal for each of the 250 measurements in the sample, and concatenate the results to form the features. We calculate the *mean, standard deviation*, and *sum* over all gain values for each signal. Additionally, we fit a line to those frequency response values and include the *slope, intercept, p-value, r-squared*, and *standard error* as features for each signal, representing each touch sample as a 250×16 matrix. The *standard error* is the average distance of all samples from the fitted regression line. *R-squared* is the percentage of the dependant variable variation that is explained by the linear model. The *p-value* is a measure of how changes in the independent variable affect the dependent variable, with a low *p-value* indicating that the independent variable is a meaningful addition to the linear regression model. This way, for every time step, we represent the system in terms of its frequency gain statistics, which we consider to be more stable, compared to the raw frequency gains.

Ultimately, we characterize a touch event as the difference between the captured touch sample represented in terms of the statistical information of its frequency gains, and the same statistical features computed over a baseline measurement where no touch occurs. The baseline is meant to capture the conditions of the surroundings, and remove their representation from the touch event characterization, so that the information that remains is more indicative of the properties of touch position. Figure 6 visualizes the feature extraction process.

3.2.2 Neural Network Model. Data processed as described above is used as an input to a neural network, which relies on three recurrent layers, followed by one layer of fully-connected perceptron consisting of one layer of 100 nodes, and an output layer with 36 nodes, as illustrated in Figure 7. Our three recurrent layers are based on LSTM cells, a stable version of RNN, which address the vanishing and exploding gradient problems faced by simple RNN cells. LSTMs have been widely used for sequential data analysis, including time-series signal data, since they capture the temporal dependencies.

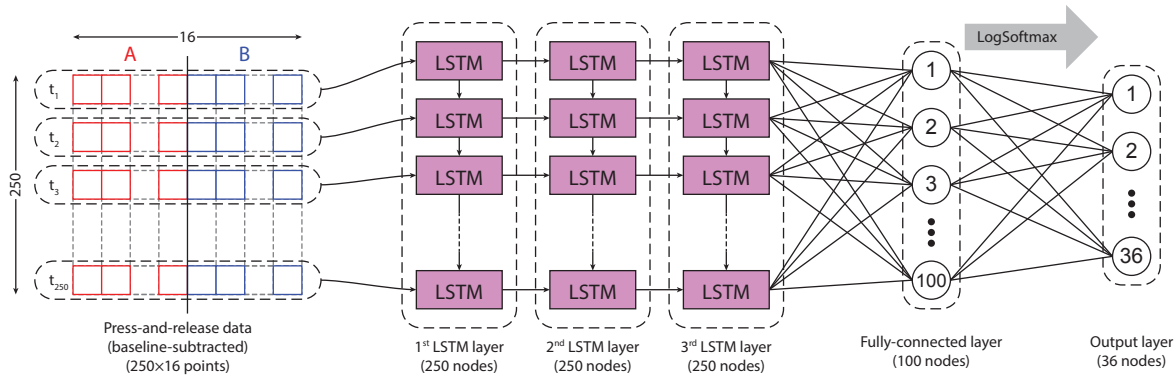


Fig. 7. LSTM Neural Network model. The network is composed of 3 subsequent layers of LSTM capturing the temporal dependencies. The last layer is linear, which then translates the input to one of the 36 button positions, through a LogSoftmax activation function.

We use the sequence of the frequency gain statistics captured from each electrode signal to construct the input to our system. Each input consists of one key-press on the touch location, represented as a sequence of 8 different statistical measures computed over values of all frequencies from each of the two electrodes. The captured sequence is 250 time steps long, with 16 values per time step. Therefore, the input layer of our neural network architecture has a size of 16. The frequency gain statistics are temporally-related, which the neural network architecture aims to capture. The size of the output layer is 36: the final linear layer maps each input into a *button* position on the knitted touchpad through a LogSoftmax activation function. The above description corresponds to the architecture of the network, however, to create an accurate model, this network needs to be trained with user data, so that its weights are adjusted accordingly. In Sections 4.1 below we describe the training and evaluation process for our model.

4 MODEL TRAINING AND EVALUATION

To move knitted sensors toward real-world use, we conducted a series of three studies of the system described above. The first study consisted of collecting user data to create a model of interaction that would allow accurate location sensing in real-time, provided the hardware infrastructure is in place for real-time sensing. We then tested the accuracy and robustness of our system by conducting two more user studies: one validation study to evaluate the performance of our model on new data, and another to test its robustness to everyday use activities. The robustness study consisted of two conditions: *(1)* proximity to a fluorescent light lamp, as a source of EMI, which as discussed above, causes a disturbance in the signal of capacitive circuits, and *(2)* stretching of the sensor, which alters its geometry. We evaluated our system's performance under each of these conditions on the trained model, resulting from the first user study. All three of these studies report *subject-independent* accuracy results: no calibration was performed for each subject. We discuss these results in a usability context, as well as compare our recognition model to other benchmark algorithmic solutions.

4.1 Study 1: Model Creation and Cross-Validation

For this study, we focused on the knitted touchpad, illustrated in Figure 4, and described in Section 3.1.1. We trained the neural network described in Section 3.2.2 to be able to classify the button that was pressed. The model

created during this study can be used in conjunction with the real time signal acquisition and processing system in order to classify location of touch.

4.1.1 Dataset Description: In order to create a trained neural network model, we collected button-press data from 24 subjects, who were college students. Each subject conducted 10-20 trials, with one trial consisting of pressing each of the 36 touch locations on the knitted pad once. There were a total of 13, 896 labelled key-press data instances: 386 samples of key-presses for each of the 36 touch locations on the keypad. Each key-press sample had a size of 250×16.

4.1.2 Methods and Resources: Data was collected and saved to a computer running the *Processing* IDE. A Processing sketch was used to visualize and save the frequency gain values from each electrode, which were returned by the Bode analysis in real-time. A Keysight MSOX3024T Oscilloscope was used to visualize the input and output waveforms in real-time. An example of data collection on a subject can be seen in Figure 1(c), and a more comprehensive data acquisition and signal processing description can be found in Section 3.1.

First, we constructed features from these raw data, as described in Section 3.2.1. The resulting representation was then used to train a neural network and tune its parameters through 10-fold cross-validation. On every iteration, all the data from 2 or 3 subjects was used for validation, while data from the other subjects was used to train the model. The dataset was balanced for all classes, since subjects were required to press on every button for each trial. The network weights were initialized using Xavier initialization [12].

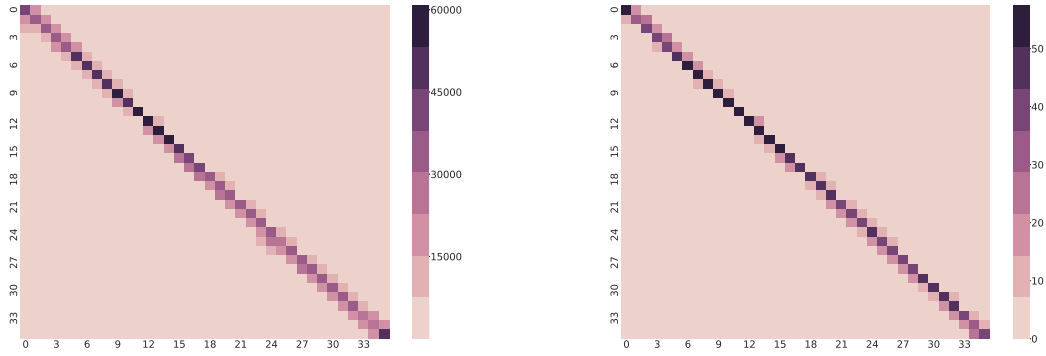
Our model was trained using PyTorch 1.4 [25] with CUDA 10.0, using a learning rate $\alpha = 0.001$, a dropout rate = 0.6 and a batch size = 128, over 2000 epochs. We used *Adam* as an optimizer, and a negative log likelihood loss function. The machine was running Ubuntu 18.04, with an Intel® Core™ i9-9960X CPU @ 3.10GHz, 128 GB RAM, and 2x RTX 2080 Ti Blowers with NVLink cards.

4.1.3 Results: To evaluate our model, we calculate the accuracy of identifying touch location on the knitted component. *Accuracy* is the ratio of correctly predicted key-presses to the total number of key-presses evaluated. As previously mentioned, there were 36 classes corresponding to 36 *buttons* or touch locations, thus, chance accuracy is 2.78%. Our model achieved an accuracy of 58.3%, which was calculated as the average validation accuracy of all 10 folds. For each fold, accuracy was calculated as the average of the achieved validation accuracies of the last 50 epochs. Table 1 shows accuracy results of all user studies, and Figure 8(a), the average classification matrix of this study, which was computed over the classification matrixes of each of the 10 models of cross-validation. We also calculate the *macro-precision*, *macro-F1-score*, and *macro-recall*. Precision refers to the ratio of true positives to the combined number of true and false positives. Recall is the ratio of true positives to the combined true positives and false negatives. The F1-score refers to the harmonic mean of precision and recall: $F1 = 2 * (precision * recall)/(precision + recall)$. Macro-precision, macro-recall, and macro-F1 refer to the balanced respective scores per class, and those values are obtained by averaging all respective class scores.

4.2 Study 2: Evaluation with New Sensor Touch Data

In order to determine the accuracy of the neural network model, we collected sensor touch data from 7 other users, whose data had not been previously seen by our model, either in the training or validation set. The purpose of this study was to evaluate the performance of the system under normal conditions—similarly to the ones when training was conducted.

4.2.1 Dataset Description: The subjects of this study were 7 college students, and the experiment set-up was the same as during the initial study, described in Section 4.1.2. Subjects pressed each of the 36 keys of the rectangular knitted sensor keypad 10 times for this evaluation, for a total of 2,520 key-presses, or 70 per button—our defined



(a) (b)

Fig. 8. Classification matrices produced with results from study 1 *(a)*, and study 2 *(b)*. The matrix rows denote the true row categories of the key-presses, while the columns show the ones predicted during evaluation. Each category is denoted by the respective button number of the knitted touchpad. A higher-value diagonal, with the rest of the matrix having lower values, would be a desirable combination indicating a high accuracy. This figure illustrates the challenges of classifying arbitrarily-defined positions on a continuous element—the buttons are sometimes classified as the ones adjacent to them.

area of touch classification. We recorded the true labels, which were the button positions on the sensor, as well as the predicted labels of our model.

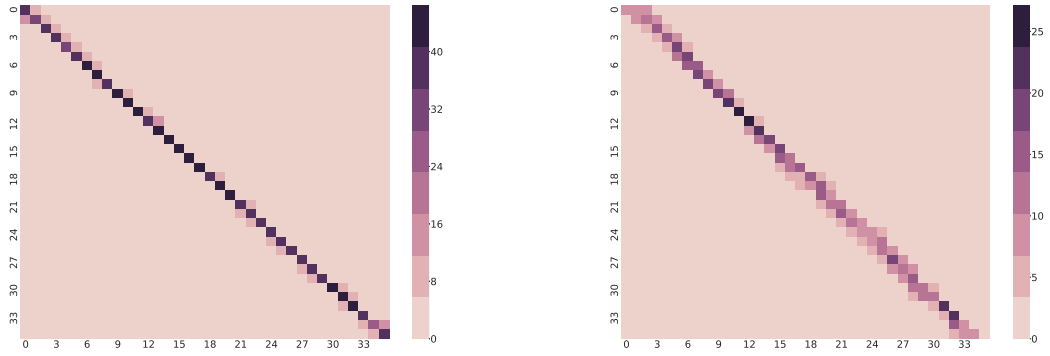
4.2.2 Methods and Resources: The collected data was used as a testing set towards the neural network models created in Section 4.1. Features were constructed on the key-press signals before they were used as inputs to the trained model for evaluation. Testing was also performed on the same machine as was training, as described in Section 4.1.

4.2.3 Results: An average accuracy of 66% was achieved during this experiment, as recorded in Table 1. This accuracy was calculated by testing the new data against each of the 10 training models created during cross-validation, and taking the average of the results. Figure 8(b) shows the corresponding classification matrix. Similarly, this classification matrix was generated as the mean of all classification matrices produced from the data tested against each trained cross-validation model. Its respective macro-precision, macro-recall, and macro-F1 scores are also reported in Table 1. These results, which are even better than the original validation results from Study 1, indicate the robustness of our model.

4.3 Study 3: Robustness to Sensor Distortion

The sensors are intended to work in everyday life, behaving as fabric, and resisting signal distortions. For this reason, we took the fabric piece on which our model was trained, and exposed it to two different types of sources that could cause signal distortion: *(1)* exposure to electromagnetic interference and *(2)* stretching.

4.3.1 Dataset Description: We recorded touch data from 4 different subjects for *condition (1)*, which consisted of exposing the sensor to EMI through a fluorescent light bulb, placed 12-18 inches away from the sensor. We also recorded data from 3 other subjects for *condition (2)*, with the sensor being stretched. Each subject pressed the



(a) (b)

Fig. 9. Classification matrices representing the accuracy of experiments in study 3 condition 1 *(a)*, and study 3 condition 2 *(b)*. The matrix rows denote the true row categories of the key-presses, while the columns show the ones predicted during evaluation. Each category is denoted by the respective button number of the knitted touchpad. Diagonal elements having relatively high values, and non-diagonal matrix elements having lower values would be a desirable combination, indicating high accuracy. Similarly to Figure 8, sometimes buttons are mistakenly classified as the ones close to them, due to the continuity of the sensing yarn.

36 buttons of the sensor 10 times. Similarly to the other two data-collection user studies, subjects were college students. For *condition (1)*, the dataset consisted of 1, 440 key-press samples; for *condition (2)*, there were 1, 080 key-press samples.

4.3.2 Methods and Resources: The methods used in this study were the same as those of study 2. For each condition, its respective dataset, after being processed through feature construction, was used to evaluate the model created in study 1. The same machine was used to run these experiments, as that in studies 1 and 2.

4.3.3 Results: The results of these experiments, are summarized in Table 1, and Figure 9, and were computed in the same way as those of study 2. An accuracy of 80% was reached for *condition 1*, while the accuracy for *condition 2* was 37%. From these results, we can see that our model retained its robustness even under conditions of signal and sensor distortion, especially during *condition (1)*, which is promising toward using it in the real-world. In *condition (2)* we see that the model does not hold as well against the structural distortion of stretching.

Table 1. Classification results for the three studies.

Study	*Accuracy*	*Macro-F1*	*Macro-Precision*	*Macro-Recall*
Model Training	0.583	0.516	0.516	0.518
Model Evaluation	0.662	0.662	0.665	0.662
Robustness: EMI	0.795	0.794	0.796	0.795
Robustness: Stretching	0.372	0.364	0.393	0.372

4.4 Benchmark Comparisons to Existing Techniques

To demonstrate that our recognition model is necessary to achieve robust location sensing, we compare the accuracy of our recognition model to three other benchmarks, which are related to our process. In particular, we report results when a single frequency Bode analysis is used (*Benchmark 1*) [36]. We further report on results when the MSD algorithm is used for feature construction, similarly to prior work [36] (*Benchmark 2*). Finally, we report on results using our same features, but a different neural network classifier (multilayer perceptron) for comparison (*Benchmark 3*). We describe these benchmarks below, and summarize their results in Table 2.

Table 2. Accuracy results of our recognition model as well as other 3 other benchmark models.

Method	*Multi-frequency?*	*Features*	*Classifier*	*Study 1*	*Study 2*	*Study 3-1*	*Study 3-2*
Benchmark 1	No	Raw	LSTM	0.253	0.256	0.313	0.203
Benchmark 2	No	MSD	LSTM	0.050	0.057	0.051	0.081
Benchmark 3	Yes	Section 3.2.1	MLP	0.463	0.435	0.560	0.258
Our model	Yes	Section 3.2.1	LSTM	**0.583**	**0.662**	**0.795**	**0.372**

In *benchmark 1* the same LSTM classifier is used, however the features are different. The purpose of this comparison is to compare the effectiveness of using our proposed feature extraction process, in conjunction with multiple frequency gains obtained from the Bode analysis. Only one frequency is input into the system, in contrast to 92 different frequencies used in our introduced method. The frequency gains are measured at each of the two sensor endpoints, as described in previous work [36]. For this benchmark method, we did not use any feature construction: the features are composed of the raw frequency values of the two endpoints, therefore each key-press in this case is represented as a 250×2 matrix. The neural network and its settings were the same as the ones we used in our method, which produced the results in Table 1. Previous work in [36] however, did not include classification. We added the LSTM step in the benchmark method for consistency, in order to make the related aspects of the two projects comparable.

Benchmark 2 uses the same frequency value as *benchmark 1* and an LSTM neural network, but it applies the MSD algorithm introduced in [36] as a feature construction step. We can see from these results that our proposed processing solution would outperform the work in [36], even if that work included the same classification step, assuming MSD was applied in this way to the singe-frequency data. MSD could potentially be applied to the signal in different ways for high accuracy, such as a feature construction step to the raw signal data of all 192 frequencies. It can also be combined with other feature extraction methods in more complex ways, including the one presented here. Alternatively, its can be integrated into a custom neural neural network. However, these explorations are beyond the scope of this work.

Benchmark 3 consists of a model which has the exact feature construction steps as the one we introduce, including multiple frequencies. The difference is in the neural network classifier. Instead of an LSTM, we use a Multi-Layer Perceptron (MLP) network, which does not necessarily retain time dependency information. All the 250 processed time instances of the key-press are concatenated and input into the network and labelled with the button position. As can be seen from Table 2, our model outperforms the benchmarks in all experiments. Below, we discuss these results more in depth.

4.5 Results Summary

While we increased accuracy of touch detection shown by cross-validation in our trained model (Study 1), and confirmed its validity in our evaluation study (Study 2), such sensors are not intended to be used in controlled lab

environments, and so we explored the model performance under conditions that would cause signal distortion (Study 3).

The results of the first three studies are summarized in Table 1, which includes average accuracy, macro-F1 scores, as well as precision and recall. These measurements values are close to each other, showing balance in the constructed model. Additionally, as can be seen from Figures 8 and 9, most cases of buttons being classified incorrectly were the ones classified as buttons positioned next to the ones pressed.

Study 3 shows that the model is mostly consistent, even when exposed to some distortion. Study 2, which included testing new data under regular conditions, shows an increased accuracy compared to the cross-validation results which indicates model stability. In Study 3, *condition 1* of the experiment reached even higher accuracy results. This indicates no visible or measurable negative effect of exposing the sensor to EMI radiation on its ability to localize touch, which is a highly desirably quality for a sensor. In Study 3, *condition 2* showed that stretching the sensor decreases its ability to accurately identify location of touch, primarily because the resistance of the individual buttons changes. In the future, we plan to address this aspect by designing algorithms with more robust spatial resolution. However, these results are still encouraging since they mean that this sensor can behave like fabric and retain some accuracy.

It is worth noting that our model was not trained with data collected under the two distortion conditions with which we experimented (EMI and stretching). We would expect the accuracy of these tests to have been higher, had our network been trained on such data. The main reason we did not, was to investigate the effect of unknown possible distortions. For a finished product implementation, training should be performed under a much larger variety of conditions, including the ones mentioned above. Moreover, the accuracy results introduced here are subject-independent to make the experience of multiple users interacting with the same sensor as intuitive and non-intrusive as possible. However, if a knitted pattern is designed to be used by a single user, calibration is likely to increase the location identification accuracy, since it removes variations that are caused by different users' physiological states.

Finally, the benchmark studies put our results in the context of other proposed methods. We showed that our features were more robust than those used in prior work [36], and also other neural network classification algorithms. In the next section, we introduce several possible design forms using one yarn and two connections, some of which can be directly plugged into the existing system to output location of touch, after individual user data training.

5 EXPLORING FURTHER DESIGN PATTERNS AND APPLICATIONS

(a) (b) (c) (d)

Fig. 10. Design patterns and applications: *(a):* The *Single-Button Game Controller* sensor controlling the *Flappy Bird* game application *(b):* The *Multi-Button Game Controller* prototype used for the *Whack-A-Mole* game; *(c):* A knitted sensing pattern of larger size than in *(a)* used for potential functionality mounted on the wall; *(d)*: *Slider Game Controller* controlling the paddle position within the *Brickbreaker* game.

We introduce different design forms and associated applications of knitted sensors, constructed using one continuous sensing yarn. The conductive yarn is knitted along a serpentine pattern, complying with the knitting process requirements (Section 2.2), and in a similar way to the sensor described in Section 3.1.1. To aid reproducibility, and support production within a manufacturing ecosystem, we use commercially available materials to construct our textile sensors: carbon-suffused nylon [20] for the conductive yarn, and polyester for the non-conductive yarn. The design process for these sensors, similarly to previous work [36, 37], includes programming a specific design pattern, and then uploading it to the weft-knitting machine, which produces each sensor according to the design. We used the *Knit Paint* program in the *Shima Seiki SDS-ONE APEX3* knitting design software suite to program the loop structure for each design. We produced the sensors using *Shima Seiki* SWG041N and SSG122SV computerized flatbed weft knitting machines [1], which are versatile in producing small to medium-sized textiles, such as shirts or gloves. We considered warp knitting and circular knitting as potential manufacturing solutions, however, flatbed weft knitting best fits the needs for rapid prototyping of fabric circuits due to the ease of setup and versatility with regards to stitch assembly. Warp knitting requires more setup time and a much more robust support infrastructure, such as yarn creels and beam warping machines, to produce textiles. Circular knitting is similar to flatbed weft knitting, but offers less flexibility for rapid prototyping and form factor.

Our knitted samples vary in pattern and dimension, and each can be used as a *standalone knitted controller* or connected in a *system of knitted components* to create interactive devices. We introduce three game controllers (Figure 10) which provide user input to different applications, and three media control sensors, which can be used individually or together to control an application, illustrated in Figure 11. These components can also be produced in a variety of sizes, colors, and materials of non-conductive yarn, as seen in Figure 1(a). We illustrate examples of patterns that have uniform sensitive areas, and others with non-uniform ones.

Although the yarn routing of each pattern differs, each circuit maintains only two connections to external sensing hardware. The embedded hardware is interchangeable between knitted sensors. The touch location decoupling method used in these proof-of-concept designs relies on the previously introduced method of *differential capacitive sensing* [37], not on the methods we develop in this work. These touch patterns are designed to explore the ability to recognize discrete position, amount of pressure applied, or continuous swipes based on their construction. The properties and design intentions behind each component are described below:

(a) (b) (c) (d)

Fig. 11. Intuitive design for knitted components that work together to control the same application. *(a)*: The Piano Keyboard. *(b)*: The Volume Slider. *(c)*: The Media Control Buttons. *(d)*: The piano keyboard, volume slider, and media control button touchpad connected in a system. The sensing hardware is networked via I2C, with a single sensing controller directing communication between all three devices. The master I2C device connects to a MacBook Air via USB MIDI to interface with *GarageBand*.

- *Single-Button Game Controller:* This prototype, illustrated in Figure 10(a), has a large circular sensing area in the middle. It can be used as a controller for games that rely on pressing a button, as well as controlling

pressure [36, 37]. This can be a desirable quality for a game controller, that can possibly enable applications current hardware controllers do not. We demonstrate its functionality by using it as a controller for the *Flappy Bird* game, which prompts the user to guide a shape through gaps in a wall, with the shape rising or falling depending on the user's applied pressure.

- *Multi-Button Game Controller:* This design, as seen in Figure 10(b), is composed of several square-shape sensing areas. The primary interaction modality for this pattern is pressing one of the large sensing locations. This prototype was designed as a controller for a *Whack-a-Mole*-style game. Touching a button when prompted by the application (flashing red) triggers an in-game hit. Such sensors can be incorporated on different surfaces for a variety of functionalities and applications (Figure 10(c)).
- *Slider Game Controller:* This pattern, illustrated in Figure 10(d) is intended to capture continuous input, such as *swiping motion*. This controller can be used for games that rely on a continuous sliding input, such as *Brickbreaker*, during which the user controls a horizontally-moving slate, aiming to catch a bouncing ball.
- *Piano Keyboard:* This prototype, demonstrated in Figure 11(a), is a 25-button touch-pad knitted to emulate a 2-octave piano instrument. The design and spacing of its keys follow those of a regular piano keyboard. It is intended to be a familiar and intuitive interface, with the added flexibility and light weight allowed by the fabric material. This prototype is an example of a knitted sensor with irregular size and spacing of buttons. The *piano keyboard* is used as a MIDI controller for the *GarageBand* application.
- *Volume Slider:* A 22-row slider, illustrated in Figure 11(b), was knitted to control the volume in *GarageBand*. It relies on a swiping gesture used to control continuous input, similarly to the *Slider Game Controller* described above, but its proportions are smaller. Both its form and the sliding mode of engagement aim to be intuitive, and conform to users' expectations of a volume controller. Rows are knitted approximately 1/8 inch apart, to maintain contact with the user's finger as it slides along the pad.
- *Media Control Buttons:* The button pad depicted in Figure 11(c) contains 3 buttons serving as discrete inputs to the same application, while graphically showing their functionalities in familiar forms: rewind, pause/play, and fast-forward.

Table 3. Detailed technical properties of sensor prototypes.

Design Form	*Size*	*Overall Resistance*	*Avg. Cross-TP Res.*	*Avg. Inter-TP Res.*
Single-Button Game Controller	36×36 in.	907 kΩ	357 kΩ	-
Multi-Button Game Controller	36×3 in.	1.75 MΩ	3.68 kΩ	213 kΩ
Slider Game Controller	36×3 in.	2.06 MΩ	42.68 kΩ	15.69 kΩ
Piano Keyboard	15×4.75 in.	783 kΩ	5.99 kΩ	24.26 kΩ
Volume Slider	6×7 in.	594 kΩ	18.53 kΩ	7.51 kΩ
Media Control Buttons	7.25×6.25 in.	206 kΩ	8.13 kΩ	47.99 kΩ

While designing these sensor geometries, it is important to abide by the knitting process specifications and requirements, detailed in Section 2.2. The serpentine routing of the yarn is a function of the knitting machine. When designing with the purpose of connecting a knitted piece to a circuit, patterns that would cause shorting of the circuit would need to be avoided. Such patterns involve crossing of conductive elements (e.g. a T-shape). When selecting the shape of the pattern, one of the decisions was also based on its function of measuring single-point input (i.e. button press) or continuous input (i.e. swiping).

The introduced samples emulate familiar interfaces, but also have layouts compatible with knitting/yarn routing. For instance, the keys on the piano keyboard are staggered to be arranged linearly. This design facilitates a translation into knitting. For each form, we also ensure that touch positions are at least a single finger's width apart, so that buttons are not pressed inadvertently. Some technical and construction details of the above-mentioned design samples are provided in Table 3. The cross-touchpoint resistance refers to the resistance measured between the start and the end of one defined touch location, such as a button. The inter-touchpoint resistance is measured between the end of one touch location and the beginning of the other.

During a knitted sensor's design phase, it is important to consider the length and resistance of the yarn pathway between contact points. Resistance of a conductor depends on its resistivity, cross-sectional area, and length. Resistivity and area of the conductor do not change throughout each sensor, since the conductive yarn is homogeneous. Resistance increases proportionally with the length of the yarn. This means that in areas where the yarn is knitted more tightly together, the resistance is higher. A large resistance limits current flow and increases the capacitors' charge time compared to a more conductive pathway. This improves measurement sensitivity given the same magnitude of induced capacitance. Figure 12 shows a comparison between the geometries of four different knitted touchpads and their respective touch localization performance. Each touch position is plotted in relation to two values: the gain to the single-frequency input response from each sensing endpoint captured through previously-introduced methods [37], without any algorithmic or machine learning approaches applied. The data was collected from a single subject pressing on each defined sensing area 1 time, and is provided for illustration purposes. It does not constitute a complete evaluation of these sensors. We are illustrating only these sensors because their primary mode of interaction is discrete touch location identification, not pressure sensitivity or continuous swiping input usable for gestures. Those interaction modalities should also be investigated in more depth in the future.

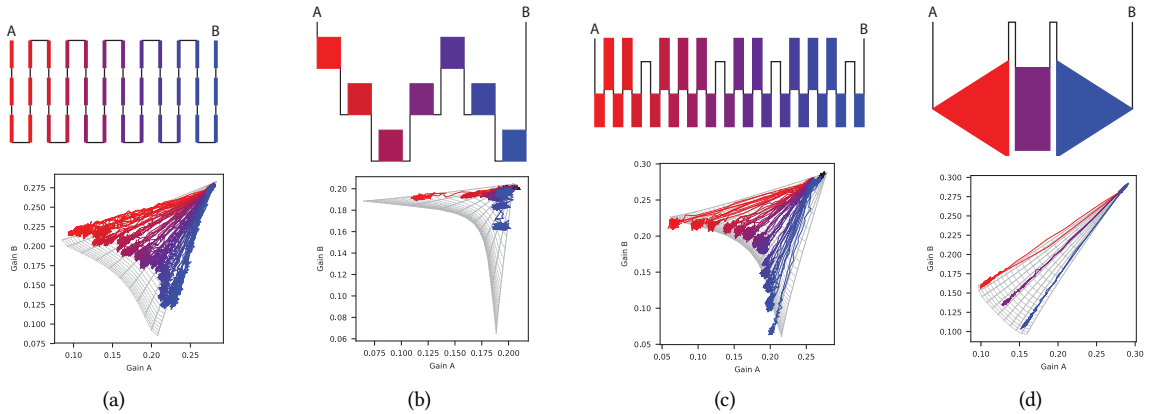


Fig. 12. Comparison of gain measurements towards touch separation for button sensors. The top right corner of each plot denotes the baseline, no-touch state. The rest of the plot shows the signal response to touch on every button location. *(a)*: The 36-button touchpad sensing area, which was used in our system pipeline and user study evaluation above. *(b)*: The Multi-Button Game Controller sensing area. *(c)*: The 25-button Piano Keyboard sensing area. *(d)*: The Media Control Buttons sensing area.

As can be seen from Figure 12, the touch location data is easier to separate into clusters even analytically, when the conductive yarn is knitted in larger areas. Figure 12(d) is an example of a relatively straightforward separation between touch location. The structure illustrated in Figure 12(a) poses a considerably greater challenge to accurate

location identification, because the sensing yarn is not assembled closely together, and its inter-button resistance is low (Table 3). In Section 3, we explored techniques to advance the capabilities of a sensor knitted with this yarn pattern, particularly because it is the most challenging of the patterns introduced here, and we predict that it would be easier to identify location of touch on the other geometries. However, the trained recognition model presented here was sensor-specific, as we have not conducted experiments with other patterns. The sensor forms presented in this section whose mode of functionality is detecting discrete locations, illustrated in Figure 12, can be plugged in to our hardware instead of the knitted touchpad. The same feature construction step and neural network architecture can also be used for touch location detection. However, each pattern needs to have its model trained with its own user data, so it can implicitly learn the position of touch for that specific geometry. In future work, we plan to explore more generalized algorithmic and machine learning solutions to enable detecting location of touch on any yarn routing path.

6 LIMITATIONS AND FUTURE WORK

The results from experiments with our sensing and recognition system, together with the variety of achievable applications and sensor configurations, demonstrate advancements towards developing extensible interactive textiles. Theoretically, any knitted component designed to detect discrete positions of touch, which is constructed using the same principles as the one knitted touchpad with which we experimented, can be connected to our real-time signal processing system. Moreover, the same feature construction and neural network architecture can be used for recognition. However, there are still several challenges to overcome, as described below.

6.1 Improving Localization Accuracy and Generalizability

Touch localization accuracy still needs to be increased in a measurable way. In addition, each model currently would need to be trained using user data specifically collected by pressing on the buttons defined by that sensor's geometry. Then, the trained model can be used to detect touch in real time. Future work could include further investigation into the shapes that are possible to construct given the constraints of knitting, as well as the optimal designs for high accuracy, taking into account considerations discussed in Section 5. More generalized circuit design and algorithmic solutions will also be explored to capture the touch location independent of yarn routing and design patterns. In this work, to explore accurate classification under various conditions, we performed the classification offline, using the data from the real-time signal acquisition and processing system. However, it is feasible to deploy a trained model to a handheld device or microcontroller, which can communicate with the rest of the system to determine location of the particular key press.

6.2 Understanding and Improving Durability in the Real World

The physical durability of the sensors must be quantified, since such sensors need to be able to withstand exposure to elements in everyday life. Additional experiments are needed to examine the versatility of the carbon-suffused nylon to material aging and abrasion. Furthermore, methods of surface enhancement and preservation, such as coating and lamination, should be investigated to quantify their effects on the physical robustness and sensing performance of the treated sensors. Moreover, the robustness of our models was evaluated for two potential distortions: stretching and exposure to fluorescent light. Another condition similar to stretching in effect is washing and drying the sensor for the first time after producing the knitted piece. Some distortion is expected to occur since the knitted fabric might shrink in size to some extent. For this reason, location identification would be more accurate if training was performed after the sensor was washed and dried first. A thorough evaluation should explore these and other factors, including exposure of the sensors to different environmental conditions and humidity, and the sensors undergoing repeated washing and drying cycles. For real-world use, these factors should not lead to loss of their conductive properties, or their ability to correctly detect touch location on an

already trained model. We plan to explore the potential of knitted sensors relying on conductive yarns with different properties.

Another aspect to be considered is sensor performance while being worn or integrated into clothing. While more data is needed to draw a conclusion, we can estimate a response based on the sensor characteristics. The sensor in proximity to the human body induces capacitance. When worn near the skin, there is an additional capacitance induced which is greater than the measured baseline parasitic capacitance. Ideally, when worn, the skin should not contact the sensor. Shielding the sensor from underside contact is currently achieved by knitting a back layer on the sensor and routing conductive yarn only on the top layer. Moving while wearing this sensor is expected to produce little to no distortion. Sweat, however is a potential source or ambiguity in location identification, since it contains electrolytes. If it seeps through the sensor, the overall resistance of the sensor will change, which is expected to affect its localization ability. Moreover, if a user is grounding himself or herself while touching a location on the sensor, his/her conductivity is increased, which will again affect the sensor response. Further studies are needed to fully investigate the extent of the effect of such conditions.

6.3 Potential Interaction Modalities and User Experience

Other modalities of interaction, enabled by the designs introduced above, such as gesture recognition and pressure sensitivity should be studied and implemented. For example, soft touch, which would cause a weak applied capacitance, could potentially pose a problem towards accurate localization. Data collected using different fingers for key-presses, different finger-placements on the sensing pad, different body poses, as well as subjects of different ages will need to be explored, due to possible changes in physiology which affect conductivity. Additionally, our results are also currently limited to single touch detecting — multi-touch detection is an area that still needs investigation, as is gesture recognition.

Usability studies need to be performed to explore the potential for adaptation of knitted sensors into end-users' everyday lives. One could also investigate new areas of possible integration for these sensors that are unique to them, and do not rely on existing hard electronic prototypes. Future work will focus on further investigating these aspects of this technology, as well as building specific applications, testing their performance and usability in real-world scenarios, getting user feedback about design aspects.

7 DISCUSSION AND CONCLUSION

Textiles evolved with humanity, from serving the function to protect and keep warm, to showing personal style, and recently to enhance experience by incorporating increased connectivity into our surroundings. In this work, we explore the potential for knitted sensing fabric to be used as a viable embedded interface.

Through three user studies, and comparisons with benchmarks, we moved towards addressing two key aspects of bringing these sensors to the real world: increased accuracy, and robustness of the system. We constructed a recognition model which uses a LSTM neural network to classify the location of touch on one of the knitted sensors with 36 touch locations. These touch locations were pre-defined from the sensor design, and a touch event was recorded in real time through our signal acquisition and processing system. The model was trained using data from 24 subjects, and achieved an average subject-independent accuracy of 58% from the different cross validation folds. Subsequently, we evaluated our system using new user touch data from 7 new users under normal circumstances, achieving an accuracy of 66% on the 36-button sensor. We also tested the system using the same previously-trained model under conditions of two possible distortions: the effects of electromagnetic interference (80% accuracy), and while the knitted section was being stretched (37% accuracy). Finally, we compared our results with several related methods as benchmarks (using only one frequency response with Bode analysis; using MSD feature construction; and using a different neural network classifier) to further demonstrate the robustness of our models.

Previous work has shown that such sensors can be produced using weft knitting, an industry-standard manufacturing technique, for easy integration into clothes and other textile surfaces. The extensible nature of weft knitting allows the sensor to conform to many shapes and the simplicity of the structure allows it to be integrated into other textiles. All the sensors introduced in this work are based on one carbon-coated conductive yarn, which is combined in the knitting process with other regular yarns to produce the fabric. Two electrodes are attached to the yarn, one on each end, to measure voltage across it. We demonstrate several design forms and related applications of knitted capacitive sensors: a *Single Button Game Controller*, a *Multi-Button Game Controller*, and a *Slider Game Controller*, each with their associated game applications; a knitted MIDI controller system, composed of a *piano*, *volume controller* and *media control buttons*.

The technical contributions and studies conducted in this work helped us lay the foundation to advancing towards viable, robust knitted textile touch interfaces.

ACKNOWLEDGMENTS

We thank Sol Schade, Keith Taylor, and Amy Stoltzfus from Drexel University's Pennsylvania Fabric Discovery Center at the Center for Functional Fabrics and Robert Lehrich from SHIMA SEIKI USA, for their invaluable digital knitting expertise and supervision of student collaborators. We thank student volunteers Nicole Chin, Colby Mills-King, and Connor Koletar for their assistance in creating the knitted MIDI controller interface. We thank student interns Laila Gaut, Curtis Patey, Katarina Galic, Lindsay Alshouse, and Joyce Dong for their assistance in creating the large knitted game pads and applications. We thank Andrew McDonald for his insight and advice regarding technical implementation strategies. This research is supported by the Pennsylvania Fabric Discovery Center and the Advanced Functional Fabrics of America grant W15QKN-16-3-0001.

REFERENCES

[1] [n.d.]. *SHIMA SEIKI USA [Online]*. https://shimaseikiusa.com/

[2] Talha Agcayazi, Michael McKnight, Hannah Kausche, Tushar Ghosh, and Alper Bozkurt. 2016. A finger touch force detection method for textile based capacitive tactile sensor arrays. In *2016 IEEE SENSORS*. 1–3. https://doi.org/10.1109/ICSENS.2016.7808528

[3] H. W. Bode. 1940. Relations between attenuation and phase in feedback amplifier design. *The Bell System Technical Journal* 19, 3 (July 1940), 421–454. https://doi.org/10.1002/j.1538-7305.1940.tb00839.x

[4] Amy Chen. 2020. The Design and Creation of Tactile Knitted E-textiles for Interactive Applications. In *Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied Interaction*. 905–909.

[5] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[6] D. De Rossi, F. Carpi, F. Lorussi, A. Mazzoldi, E. P. Scilingo, and A. Tognetti. 2002. Electroactive fabrics for distributed, conformable and interactive systems. In *SENSORS, 2002 IEEE*, Vol. 2. 1608–1613. https://doi.org/10.1109/ICSENS.2002.1037364

[7] Laura Devendorf, Joanne Lo, Noura Howell, Jung Lin Lee, Nan-Wei Gong, M Emre Karagozler, Shiho Fukuhara, Ivan Poupyrev, Eric Paulos, and Kimiko Ryokai. 2016. " I don't Want to Wear a Screen" Probing Perceptions of and Possibilities for Dynamic Displays on Clothing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 6028–6039.

[8] Rachel Freire, Cedric Honnet, and Paul Strohmeier. 2017. Second Skin: An Exploration of eTextile Stretch Circuits on the Body. In *Proceedings of the Eleventh International Conference on Tangible, Embedded, and Embodied Interaction*. 653–658.

[9] Francine Gemperle, Chris Kasabach, John Stivoric, Malcolm Bauer, and Richard Martin. 1998. Design for wearability. In *digest of papers. Second international symposium on wearable computers (cat. No. 98EX215)*. IEEE, 116–122.

[10] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 1999. Learning to forget: Continual prediction with LSTM. (1999).

[11] Scott Gilliland, Nicholas Komor, Thad Starner, and Clint Zeagler. 2010. The Textile Interface Swatchbook: Creating graphical user interface-like widgets with conductive embroidery. In *Proceedings of the 2010 International Symposium on Wearable Computers (ISWC '10')*. 1–8. https://doi.org/10.1109/ISWC.2010.5665876

[12] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 249–256.

[13] Nur Al-huda Hamdan, Jeffrey R. Blum, Florian Heller, Ravi Kanth Kosuru, and Jan Borchers. 2016. Grabbing at an Angle: Menu Selection for Fabric Interfaces. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers (ISWC '16)*. ACM, New York, NY, USA, 1–7. https://doi.org/10.1145/2971763.2971786

[14] Nur Al-huda Hamdan, Simon Voelker, and Jan Borchers. 2018. Sketch&Stitch: Interactive Embroidery for E-textiles. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. ACM, New York, NY, USA, Article 82, 13 pages. https://doi.org/10.1145/3173574.3173656

[15] Yuji Hasegawa, M. Shikida, D. Ogura, and Kazuo Sato. 2007. Novel type of fabric tactile sensor made from artificial hollow fiber. In *2007 IEEE 20th International Conference on Micro Electro Mechanical Systems (MEMS)*. 603–606. https://doi.org/10.1109/MEMSYS.2007.4433079

[16] Cedric Honnet, Hannah Perner-Wilson, Marc Teyssier, Bruno Fruchard, Jürgen Steimle, Ana C Baptista, and Paul Strohmeier. 2020. PolySense: Augmenting Textiles with Electrical Functionality using In-Situ Polymerization. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[17] Dana Hughes, Halley Profita, and Nikolaus Correll. 2014. SwitchBack: An On-body RF-based Gesture Input Device. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers (ISWC '14)*. ACM, New York, NY, USA, 63–66. https://doi.org/10.1145/2634317.2634343

[18] Dana Hughes, Halley Profita, Sarah Radzihovsky, and Nikolaus Correll. 2017. Intelligent RF-Based Gesture Input Devices Implemented Using e-Textiles. (2017). https://doi.org/10.3390/s17020219

[19] Masayuki Inaba, Yukiko Hoshino, K. Nagasaka, T. Ninomiya, Satoshi Kagami, and H. Inoue. 1996. A full-body tactile sensor suit using electrically conductive fabric and strings. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96*, Vol. 2. 450–457. https://doi.org/10.1109/IROS.1996.570816

[20] Jarden Applied Materials 2014. *RESISTAT® TYPE F901, MERGE D044*. Jarden Applied Materials. http://www.resistat.com/pdf/f901d044color.pdf

[21] Thorsten Karrer, Moritz Wittenhagen, Leonhard Lichtschlag, Florian Heller, and Jan Borchers. 2011. Pinstripe: eyes-free continuous input on interactive clothing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 1313–1322.

[22] Keil. [n.d.]. ARM CMSIS-DSP. https://arm-software.github.io/CMSIS_5/DSP/html/modules.html.

[23] Simon Ozbek, Md Tahmidul Islam Molla, Crystal Compton, and Brad Holschuh. 2018. Novel manufacturing of advanced smart garments: knitting with spatially-varying, multi-material monofilament. In *Proceedings of the 2018 ACM International Symposium on Wearable Computers*. 120–127.

[24] Patrick Parzer, Florian Perteneder, Kathrin Probst, Christian Rendl, Joanne Leong, Sarah Schuetz, Anita Vogl, Reinhard Schwoediauer, Martin Kaltenbrunner, Siegfried Bauer, and Michael Haller. 2018. RESi: A Highly Flexible, Pressure-Sensitive, Imperceptible Textile Interface Based on Resistive Yarns. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology (UIST '18)*. ACM, New York, NY, USA, 745–756. https://doi.org/10.1145/3242587.3242664

[25] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[26] Andreas Pointner, Thomas Preindl, Sara Mlakar, Roland Aigner, and Michael Haller. 2020. Knitted RESi: A Highly Flexible, Force-Sensitive Knitted Textile Based on Resistive Yarns. In *ACM SIGGRAPH 2020 Emerging Technologies*. 1–2.

[27] Andreas Pointner, Thomas Preindl, Sara Mlakar, Roland Aigner, and Michael Haller. 2020. Knitted RESi: A Highly Flexible, Force-Sensitive Knitted Textile Based on Resistive Yarns. In *ACM SIGGRAPH 2020 Emerging Technologies (SIGGRAPH '20)*. Association for Computing Machinery, New York, NY, USA, Article 21, 2 pages. https://doi.org/10.1145/3388534.3407292

[28] E. R. Post, M. Orth, P. R. Russo, and N. Gershenfeld. 2000. E-broidery: Design and Fabrication of Textile-based Computing. *IBM Syst. J.* 39, 3-4 (July 2000), 840–860. https://doi.org/10.1147/sj.393.0840

[29] Ivan Poupyrev, Nan-Wei Gong, Shiho Fukuhara, Mustafa Emre Karagozler, Carsten Schwesig, and Karen E. Robinson. 2016. Project Jacquard: Interactive Digital Textiles at Scale. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 4216–4227. https://doi.org/10.1145/2858036.2858176

[30] Casey Reas and Ben Fry. 2006. Processing: Programming for the Media Arts. *AI Soc.* 20, 4 (Aug. 2006), 526–538. https://doi.org/10.1007/s00146-006-0050-9

[31] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. 1988. Learning representations by back-propagating errors. *Cognitive modeling* 5, 3 (1988), 1.

[32] Stefan Schneegass and Alexandra Voit. 2016. GestureSleeve: Using Touch Sensitive Fabrics for Gestural Input on the Forearm for Controlling Smartwatches. In *Proceedings of the 2016 ACM International Symposium on Wearable Computers (ISWC '16)*. ACM, New York, NY, USA, 108–115. https://doi.org/10.1145/2971763.2971797

[33] Ryan Seguine et al. 2007. Capacitive Sensing Techniques and Considerations. *Cypress Semiconductor Corporation Mobile Handset DesignLine* (2007).

[34] Mathias Sundholm, Jingyuan Cheng, Bo Zhou, Akash Sethi, and Paul Lukowicz. 2014. Smart-mat: Recognizing and counting gym exercises with low-cost resistive pressure sensing matrix. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. 373–382.

[35] Seiichi Takamatsu, Takeshi Kobayashi, Nobuhisa Shibayama, Koji Miyake, and Toshihiro Itoh. 2011. Meter-scale surface capacitive type of touch sensors fabricated by weaving conductive-polymer-coated fibers. In *2011 Symposium on Design, Test, Integration Packaging of MEMS/MOEMS (DTIP)*. 142–147.

[36] Richard Vallett, Denisa Qori McDonald, Genevieve Dion, Youngmoo Kim, and Ali Shokoufandeh. 2020. Toward Accurate Sensing with Knitted Fabric: Applications and Technical Considerations. 4, EICS (2020). https://doi.org/10.1145/3394981

[37] Richard Vallett, Ryan Young, Chelsea Knittel, Youngmoo Kim, and Geneviève Dion. 2016. Development of a Carbon Fiber Knitted Capacitive Touch Sensor. *MRS Advances* 1, 38 (2016), 2641–2651. https://doi.org/10.1557/adv.2016.498

[38] Irmandy Wicaksono and Joseph Paradiso. 2020. KnittedKeyboard: Digital Knitting of Electronic Textile Musical Controllers.

[39] Raphael Wimmer and Patrick Baudisch. 2011. Modular and Deformable Touch-sensitive Surfaces Based on Time Domain Reflectometry. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology (UIST '11)*. ACM, New York, NY, USA, 517–526. https://doi.org/10.1145/2047196.2047264