



HandSense: Capacitive coupling-based Dynamic, Micro Finger Gesture Recognition

Viet Nguyen*

WINLAB, Rutgers University
North Brunswick, New Jersey
vietnh@winlab.rutgers.edu

Siddharth Rupavatharam*

WINLAB, Rutgers University
North Brunswick, New Jersey
sidr@winlab.rutgers.edu

Luyang Liu

WINLAB, Rutgers University
North Brunswick, New Jersey
luyang@winlab.rutgers.edu

Richard Howard

WINLAB, Rutgers University
North Brunswick, New Jersey
reh@winlab.rutgers.edu

Marco Gruteser

WINLAB, Rutgers University
North Brunswick, New Jersey
gruteser@winlab.rutgers.edu

ABSTRACT

Head-mounted devices (HMD) for Augmented Reality (AR) are gaining traction thanks to a growing number of applications in the areas of image guided therapy, computer aided design, cargo packing, manufacturing and digital field service. However, providing an always available, intuitive and user friendly input for these devices remains a challenging problem. This paper explores recognizing dynamic, micro finger gestures using capacitive coupling for interacting with a head-mounted device. Electrodes are attached to fingertips of users gloves and capacitive coupling among all pairs of electrodes is measured quickly to infer the real-time spatial relationship between fingers. The system is able to recognize fine, low-effort finger gestures, such as swiping, sliding, tap, double-tap. We evaluated our prototype with 14 gestures executed by 10 subjects and found a 97% accuracy of gesture recognition.

CCS CONCEPTS

- Human-centered computing → Interaction design; Ubiquitous and mobile computing;
- Computer systems organization → Sensors and actuators.

KEYWORDS

Human Computer Interaction (HCI); Capacitive Sensing; Gesture Recognition

ACM Reference Format:

Viet Nguyen, Siddharth Rupavatharam, Luyang Liu, Richard Howard, and Marco Gruteser. 2019. HandSense: Capacitive coupling-based Dynamic, Micro Finger Gesture Recognition . In *The 17th ACM Conference on Embedded Networked Sensor Systems (SenSys '19), November 10–13, 2019, New York, NY, USA*. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3356250.3360040>

*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SenSys'19, November 2019,

© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-6950-3/19/11...\$15.00
<https://doi.org/10.1145/3356250.3360040>

1 INTRODUCTION

Head-mounted devices (HMDs) for Augmented Reality (AR) are transforming modern workspaces thanks to their ability to overlay digital information onto the physical world. There are a growing number of applications of these devices in different industries, such as image-guided therapy [1], site productivity improvement for construction workers [2], online support for field service workers [3], training new employees [4]. However, providing inputs to these devices while being user friendly, intuitive and ensuring an immersive experience remains a challenging problem: the current input techniques mostly require users to hold a tablet or smart phone in one hand or both hands or require hands to be present in the field of view of a sensor. This often leads to inconvenient interactions and limits the device usage in mobile scenarios. For example, camera-based detection of in-air gesture interfaces (Microsoft HoloLens [5]) requires users to raise a hand to eye level, which can cause fatigue over longer periods of use and is also impractical in some scenarios, such as repair and maintenance. Voice input can be convenient for some simple instructions or information, but can be disturbing in a common workplace setting. Therefore, to advance the usage of the head-mounted devices, an *always-available, low-effort, and expressive* input method is required.

Input methods using hand or finger gestures can satisfy this need. Current techniques being used for hand/finger gesture recognition include off-body sensing (cameras [5], radar [6], acoustic [7]) and on-body sensing (inertial sensors [8], impedance tomography [9], magnetic sensors [10]). Some approaches seek to reconstruct arbitrary hand poses, but generally rely on cameras, which require the hand in the field of view and significant computational overhead, or visible light sensing [11], which also requires user hands to be inside the sensing space. In addition, gestures being recognized often include large movements of the fingers or the whole hand, which can be tiring to users during/after long periods of usage. The existing gesture recognition techniques fall short of satisfying the needs for controlling HMDs in working environments because of the following reasons: unable to operate outside a specific region of sensor operation; heavy instrumentation on the hand or in off-body sensors; difficult to detect low-effort finger gestures that are more suitable for HMD controller.

In this paper, we propose HandSense, a *light-weight, always-available* system to recognize a series of *dynamic, micro* finger

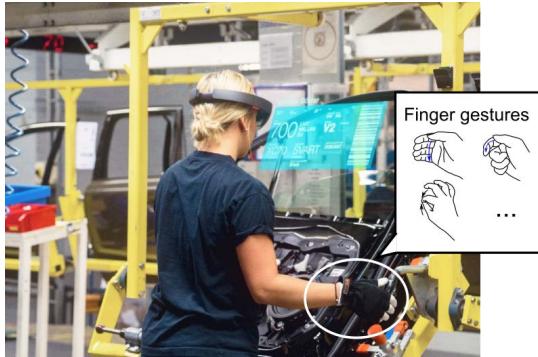


Figure 1: HandSense concept. While Augmented Reality head-mounted devices start to find applications in areas like manufacturing, repair and maintenance, providing inputs for these devices with low-effort from users remains a challenge. HandSense offers an always-available, user-friendly dynamic, micro finger gesture recognition system for these devices.

gestures that are highly suitable for controlling HMDs. The key idea in HandSense is measuring and classifying pairwise profiles of capacitive coupling between electrodes placed on all fingertips. Capacitive coupling between two electrodes is a monotonic function of the distance between them; it therefore allows inferring distance between two corresponding fingertips. Given the structural constraints of the human hand, the inferred fingertip distances allow recognizing micro finger gestures.

This approach is motivated by the observation that there exists a large and important class of augmented reality applications where users typically wear gloves (e.g., in the medical, maintenance / repair, manufacturing, or certain e-sport domains). The electrodes can be integrated into the fingertip sections of such gloves, akin to how many gloves already contain conductive materials at the fingertip to enable touchscreen use. Placing electrodes on each fingertip can therefore be much less intrusive than one might initially assume. Note also, that in contrast to more heavily sensor-instrumented gloves for sensing hand motions, such as DataGlove [12] or fiber-optic gloves for VR applications [13], HandSense only requires electrodes as sensing elements, which can be fashioned from cheap conductive materials such as copper tapes or conductive thread (connecting to an external processing unit possibly placed inside user's smartwatch or a wristband), thus the gloves can be particularly useful in medical or high wear and tear working environments. While currently intended for gloves, advances in skin electronics [14] (perhaps electrodes and traces back to the on-wrist device) may allow HandSense techniques to be used even in applications where users do not wear gloves. Overall, note that electrodes and traces are not active components, thus the fabrication can be low-cost.

Another aspect that helps HandSense better serve as a gesture controller for HMDs is its low-effort, always-available property. Since the system relies only on interactions between sensing elements on fingers, it is not limited by the working range or suffered from occlusion from external sensors (e.g. cameras [5], radar chip [6]). In addition, HMD users in working environments often have their hands occupied (e.g. therapists working on medical devices, cargo

workers holding packages); in these cases performing finger gestures with small movements in any place is the more preferred method over whole-hand movement onto the virtual dashboard, which is inconvenient and interrupting to the workflow.

There are several challenges in realizing the HandSense system. *First*, the human hand is a large conductive surface, thus the dominant capacitive coupling of the fingertip electrodes is through the hand and the signal is much less dependent on the relative distances between the electrodes. To further increase the dynamic range of the detection of spatial relationship between electrodes, we seek to reduce the unwanted influence of the hand through the use of an additional *ground electrode* on each finger. *Second*, to be able to detect quick, dynamic, micro finger gestures, the capacitive coupling measurements should be fast to provide frames of link measurements quickly. We use *synchronous undersampling* technique, which is a light-weight, low complexity method for estimating the received signal amplitude. *Third*, as over-the-air capacitive coupling between finger electrodes decreases quickly with distances, the link measurements between non-adjacent fingers are less usable in the capacitive profile. We identified an additional *through the hand* capacitive coupling path between all fingers, thus enabling more types of finger gestures to be recognized.

In summary, the major contributions of this paper are as follows:

- Proposing a placement configuration for electrodes on fingertips to enable measurement of capacitive coupling between each pair of fingers with minimal effect from user's hand.
- Designing a light-weight capacitive profiling system for measuring pair-wise capacitive coupling between fingers, which are then used for finger gesture classification system.
- Identifying three types of finger interactions detected by the capacitive profiling system, which enable more dynamic, micro finger gestures to be recognized.
- Designing and implementing a glove prototype and evaluating HandSense in recognizing a set of 14 different micro finger gestures based on data collected from 10 subjects.

2 RELATED WORK AND BACKGROUND

The advent of HMDs for Augmented Reality starts to bring to modern workplace intriguing applications, where information being overlaid on physical world in workers' views exposes new insights and improves their productivity. Existing modalities of interacting with devices, including keyboards, mice, track-pads and touch screens, are not suitable for HMDs to smoothly blend workers' experience with HMDs with their normal workflow. Input methods using hand or finger gestures are better candidates for these new devices, as they are less likely to interfere with the workflows and require less effort from users. In this section, we review related work on hand or finger gesture recognition techniques, both hand-instrumented and hand instrumentation-free approaches, then present an overview of the advantages our proposed system, HandSense, has over these techniques. The background of capacitive coupling, which is the sensing technique used in HandSense, is also discussed.

2.1 Existing finger gesture recognition techniques

Hand-instrumented approach. Data gloves are a reliable way of sensing hand gestures and movements, and have been in use since

the early seventies. Early gloves such as the Sayre Glove [13] used optical fibers with an LED on one end and a photoreceiver on the other. Bending a finger bent the optical fiber reducing the amount of received light and provided information about the gesture made. Gloves such as the Data Glove [12] and Digital data entry glove [15] focused on sensing the amount of finger flexion using bend and flex sensors to infer hand and finger gestures. More advanced gloves such as the AcceleGlove [16] and CyberGlove [17] were sensor dense and usually had a mix of flex or bend sensors, touch sensors, inertial motion sensors, tilt sensors, ultrasonic sensors, LED and photosensors sensors, with sensors mostly affixed to the glove. AcceleGlove [16] was equipped with inertial sensors (accelerometer and gyroscope) which measured roll, pitch, yaw and provided absolute angular position to help reconstruct the exact posture of the hand. While this class of "data gloves" were rich in providing data generated from different parts of the hand, the sensors were not cheap and due to their large numbers the gloves were bulky, cumbersome to carry and usually restricted the movement of the users hands. Further, the sensors could not translate small changes in flexion to finer or dynamic gestures and they also usually required a user-specific calibration procedure.

Magnetic field sensing is another technique used to track the position of fingers. Chouhan et al. [18] affix a strong magnet to the palm and Hall sensors on fingertips. uTrack [19] users wear a pair of magnetometers on the back of their fingers and a permanent magnet affixed to the back of the thumb and Finexus [10] is able to track precise motion of multiple fingertips by instrumenting the fingertips with electromagnets. All of these methods use a source element that creates a magnetic field and a small sensor that reports its position and orientation with respect to this magnetic source by measuring change in magnetic field when the fingers are moved. Though these methods of sensing do not require line-of-sight between the source and sensor, the entire system is often clunky and heavily dependent on the range of magnetic field.

Electric field sensing can also be used to measure change in distance. CapBand [20] and GestureWrist [21] detect changes in wrist contour by measuring the capacitance between a series of electrodes integrated into a wristband. Electrical Impedance Tomography [22] is a similar paradigm employed for hand gesture recognition. Tomo [23] and Touche' [24] recover the inner impedance distribution of the wrist and forearm using pair-wise measurements from surface electrodes surrounding them. The surrounding electrodes measure changes in muscle tension to detect performed gestures. Hence, gestures need to be performed with some effort to be detected. Also these techniques find it hard to detect dynamic and fast gestures.

Hand-instrumentation free approach. Cameras can be used to capture hand movements directly to infer the gestures being performed. Several systems such as the HoloLens [5], DepthTouch [25], 6D Hands [26], Keskin et al. [27] and Microsoft Kinect [28] capture raw images or video of the hand and process them using sophisticated computer vision algorithms to determine hand position and gestures being made. While camera-based techniques do not require the user to wear or carry extra devices, thus making the experience of using the system immersive and intuitive, they often raise privacy concerns and come with large computational overheads. For gesture recognition, they also require large datasets for reliable classification. These approaches also assume that the hands are always in the field

of view of the camera (i.e., no occlusion) and external illumination conditions will always permit capturing data of satisfactory quality. Moreover, integrating camera modules leads to bulky systems.

Acoustic systems such as the FingerIO [7] and SoundTrak [29] transform a device (smart-phone[7] or ring [29]) into an active sonar system that transmits inaudible sound signals and tracks the echoes from fingers at its microphones. These systems rely on line-of-sight between the source and the microphones and the surface surrounding the transmitter and receiver greatly influence reflections.

Light based technique Aili [30] uses a table lamp and few low-cost photoreceivers to reconstruct a 3D hand skeleton in real time. ZeroTouch [31] makes use of infrared LEDs and sensors for hand pose sensing. While ZeroTouch only tracks fingers in a 2D plane, Aili reconstructs 3D hand poses. SensIR [32] detects hand gestures with a wearable bracelet using infrared transmission and reflection. However, these solutions find it hard to translate small changes in flexion to micro and dynamic gestures. They also require the hands to always be in the field of view of the sensor.

Other WiFi based techniques used to perform hand gesture recognition are WiFi channel state information (CSI) [33], WiFi received signal strength (RSSI) [34] and more recently radar based systems [6]. WiGest [34] leverages changes in WiFi signal strength to sense in-air hand gestures around the user's mobile device. Whereas, WiG [33] attempts to achieve a fine-grained gesture recognition only by observing abnormalities in CSI. Similarly, SignFi [35] recognizes upto 276 sign language hand gestures using CSI. WiDraw [36] harnesses the angle-of-arrival values of incoming wireless signals at the mobile device to track the user's hand trajectory. More recently, Google's Project Soli [6] proposed a 60GHz radar chip and based on the principles of radar sensing to detect micro movements of fingers. Similar to camera and light based techniques, WiFi and radar based sensors require line of sight between transmitter and the hand. Further, these techniques also necessitate proximity to transmitters and receivers.

HandSense approach. In terms of form factors, our proposed system HandSense is closest to the hand-instrumented systems based on gloves described above. However, due to the heavily instrumented sensors used to collect useful data, these systems tend to be expensive, bulky and cumbersome for users to wear. Other systems based on more lightly instrumented form factors, such as wristbands, are not able to sense micro gestures or dynamic movements of fingers. The hand-instrumentation free approaches (e.g. cameras, radars, acoustic sensing, etc.) also have their own limitations: they require the surrounding environment to be stable, well illuminated, they expect the hands to be in the field of view and they also have a high computation overhead. The limitations of the above approaches make them difficult to be used as hand gesture control methods for HMDs in working environments, where workers have their hands usually occupied, and/or move freely in space.

HandSense is able to overcome these problems by making use of capacitive sensing. As demonstrated in the next sections, this sensing modality has the following advantages: 1) HandSense is *always available*; it infers the relative spatial relationship between fingers, hence, the hands can be anywhere independent of field of view and gestures can still be recognized, 2) by its sensitivity to close range movements capacitive sensing allows recognizing micro

finger gestures, 3) sensing electrodes are cheap and the glove is light-weight even with embedded electrodes in it, and 4) low computation overhead.

2.2 Capacitive sensing

Capacitive sensing is an ubiquitous sensing technology in human-computer interaction. It works by measuring the capacitance variation between two or more conductors. In the most basic form, the capacitance between two parallel plate conductors is $C = \frac{\epsilon_0 \epsilon_r A}{d}$, where ϵ_0 , ϵ_r are the free space and relative dielectric constants, respectively, A is the area of the conductor plate and d is the distance between the two conductors. While there are many forms of capacitors, the capacitive coupling between two conductors is always affected by only these three factors: electrode size, dielectric between electrodes and distance between them.

The measurement technique in HandSense is closely related to the shunt-mode capacitive sensing [37]. In this mode, a capacitive link is established between two electrodes, in which one electrode is powered by an AC signal and the displacement current is measured at the other electrode. The displacement current is proportional to the capacitive coupling amount between the two electrodes. Each sensing electrode can be configured as either a transmitter or a receiver. For n electrodes, there are $\frac{n(n-1)}{2}$ distinct measurements for all transmitter-receiver combinations. Note that the electrodes are in fixed positions with careful calibration to better detect the appearance/position of human body parts.

While also using excitation-response measurement approach as in the shunt-mode method, the electrodes in HandSense are placed at *mobile* positions; in particular on fingertips. It then uses the pairwise capacitive coupling measurements between electrodes to infer micro gestures performed by users. Measurement with this particular electrode placement presents both challenges and opportunities: on one hand, it is difficult to calibrate the measurement system with mobile electrodes, and the large surface of the user's hand causes most capacitive coupling between electrodes to pass through the hand. On the other hand, distance between fingers when performing gestures is small enough for a capacitive coupling measurement to work. Also, the relative motion between fingers is constrained (fingers can only flex and move in certain directions). HandSense optimizes the electrode placement to only expose the capacitive coupling path associated with finger gestures, including close-range over-the-air coupling and *intended* through-the-hand finger communication, while reducing the *unwanted* coupling in the back channel between electrodes.

3 HANDSENSE OVERVIEW

HandSense is able to recognize a set of dynamic, micro finger gestures that are suitable for use in conjunction with head-mounted devices. This is enabled by placing electrodes on each of the five fingertips of a hand and inferring the spatial relationship between them through capacitive measurements from all the pairs of electrodes. HandSense, therefore, is self-contained: unlike approaches using cameras, on-body or external RF sensors, HandSense is able to detect finger gestures when the hand moves anywhere in space, even when it is not in the field of view (FoV) of a head-mounted devices

or hands are occluded. The system is also able to detect fast movements (comparable to the speed of a quick swipe), thus allowing the gestures to be low-effort to users. The availability everywhere and the ability to detect fast, low-effort gestures make HandSense a highly suitable input method for head-mounted devices.

On-finger electrode design. A simple method to infer close distance / proximity between two electrodes in free-space is by measuring the capacitance between them as capacitance is inversely proportional to the distance between electrodes. However, a naive configuration of affixing one electrode on each finger comes with problem: a large amount of the coupling between the electrodes would be through the hand as opposed to over the air. This is because the hand is more conductive than air and most of the capacitance coupling between the two electrodes would be through the lower impedance path along the hand. Hence it becomes difficult to measure the small change in capacitive coupling through the air on top of a large capacitive coupling through the hand when the fingers move closer or further away from each other. To solve this problem, we propose adding a ground electrode underneath each signal electrode to minimize the capacitive coupling between the signal electrode and the user's hand. More discussions about this design is in Section 4.

Minimally instrumented glove design. HandSense consists of a central controller board worn on a user's wrist and a glove which is used to equip the user's fingertips with sensing electrodes. Note that the glove only requires passive components; electrodes and traces. This makes HandSense particularly useful in high wear and tear environments, such as healthcare, wellness and fitness, automobile/factory shop floor, assembly line. A user can connect their own smartwatch/wristband with a new glove to use with his head-mounted device.

Light-weight pair-wise capacitive coupling measurement techniques. HandSense is based on the insight that most finger gestures can be inferred from a profile of pair-wise capacitive coupling measurements between fingertip electrodes. Furthermore, since HandSense seeks to recognize dynamic finger gestures, the pair-wise capacitive coupling profile contains not only measurements at one instant in time but a time series of measurements, providing richer data for finger gesture classification. For typical dynamic, micro finger gestures (e.g. sliding, tapping), which can last under 1 second, the measurement system should repeatedly sample all finger-pairs fast enough to deliver sufficient data points to infer the gestures. We employ several techniques to satisfy this requirement: (a) fast switching between electrodes to act as transmitters and receivers, (b) a *synchronous undersampling* measurement technique to quickly estimate the instantaneous received signal in each link. The synchronous undersampling technique is light-weight in both hardware and firmware: it avoids the needs of expensive components such as mixers, phase shifters, and low pass filters, as in the traditional synchronous detection technique. The on-board firmware only requires a moderate ADC sampling rate and minimal computational overhead, as opposed to the Discrete Fourier Transform technique. Such low requirements made it easier to integrate the controller electronics into low-cost wristbands for HMD users. We describe these techniques in more detail in Section 5.

Finger gesture recognition. With the above electrode placements and measurement techniques, we describe three different

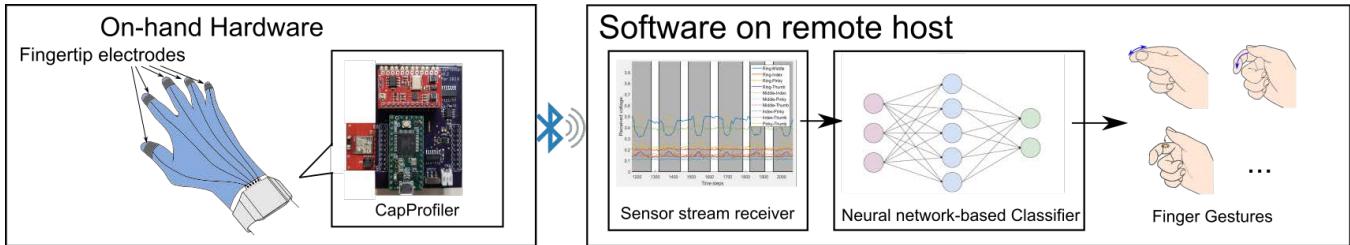
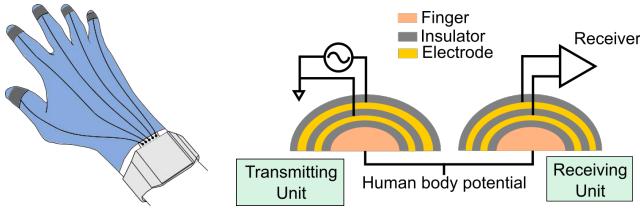


Figure 2: System overview



(a) A glove with electrodes attached on fingertips and connected with a wrist-worn device.

Figure 3: Electrode placement.

finger interactions that HandSense can recognize: direct over-the-air finger proximity, finger touching, and indirect through-the-hand electrode communication. These finger interactions produce signal signature in the time series data, which can be used for recognizing more finger gestures. With this time series data of measurements on pair-wise links, we investigate different neural network based techniques to classify the finger gestures. More details are in Section 6.

Design overview. Fig. 2 illustrates the overall design of HandSense. The sensing electrodes are attached on a glove at the fingertip sections. These electrodes are connected with a central controller board, called *CapProfiler*, which could be embedded inside a smart-watch or wristband. Inside this board, a microcontroller controls the signal transmission through a signal generator module, receives signal from an analog receiver circuit, and coordinates timing of different transmitter-receiver links. Received signal amplitudes calculated from the measured signal on all links are packaged into frames and sent over Bluetooth to a remote host, which can be a head-mounted device the user is wearing. The time-series signal sequences of all the communication links are then passed through a trained end-to-end neural network model to classify into different finger gestures.

4 DESIGN OF ON-GLOVE ELECTRODES

The electrodes (conductor plates) act as both transmitting and receiving elements in HandSense. They are placed on the top bone (distal phalanges) of each finger (Fig. 3(a)). The rationale for placing the sensing electrodes in these positions is that the fingertips are the most active parts of the hand, and they take part in almost all gestures. While we seek to measure the capacitive coupling between each pair of electrodes over the air (small dielectric), the higher dielectric constant capacitive coupling path through the hand presents a challenge to HandSense. In addition to acting like a resistor, the

outermost layer of skin (epidermis) acts like a capacitor if placed in contact with a piece of metal. The underlying tissue represents one plate of a capacitor and the metal surface the other. The dry epidermis represents the less conductive material or "dielectric" in between. In our case we use an AC source to excite the electrodes, this AC source "shorts" out the natural resistance of the epidermis allowing the current to bypass that part of the hand's resistance and making the hand's total resistance much lower. This resistance further reduces with increasing frequency of the current. This means that the dominant signal path goes through the hand (the less resistive path) as opposed to through the air. According to the National Institute for Occupational Safety and Health (NIOSH) the resistance offered by the human body is in the range of 1000 to 100,000 Ω [38] and the capacitor with $A = 2\text{cm}^2$, $d = 3\text{cm}$ with air as dielectric has a capacitance of 590pF and an impedance of 26M Ω [39]. Hence a weaker amount of capacitive coupling over the air between two signal electrodes in the presence of a stronger capacitive coupling through the body would be more difficult to measure. This is detrimental to our system as we wish to estimate the over-the-air distance between fingertips based on the capacitance between the fingertip electrodes.

To address this challenge, we place a ground electrode between each signal electrode and the finger, with insulation layers in between to prevent shorting of the electrodes (Fig. 3(b)). It is evident from Fig. 4 that there is not much change in amplitude at 100kHz when the ground plane is absent, whereas in the case with the ground plane we can see that there is a drop in amplitude when fingers are moved apart to a distance of 3cm.

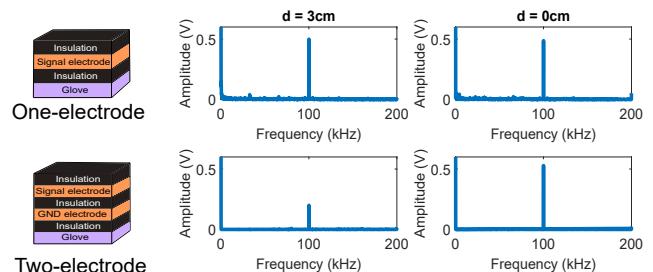


Figure 4: Received signal at 100kHz in one-electrode vs. two-electrode designs. Here d is the distance between the two fingers during its transmitting-receiving session.

Adding a ground electrode underneath the signal electrode closer to the signal electrode than the skin helps, as it is at a lower potential than the skin. Hence it couples stronger with the signal electrode. It

also provides a common ground to the smart watch/device which measures the voltage. Without the ground electrode, the transmitting signal would couple to the user's hand and then couple to the receiving electrode.

Note that there are other advantages in having a defined ground electrode: 1) the ground plane ensures that the signal is always coupled to the same ground potential across all fingertips as each fingertip has the same ground electrode underneath the signal electrode. Without this common ground, each signal electrode is coupled to its own dynamically changing finger potential, 2) generally frequency multiplexing (i.e., each finger is assigned a pre-determined frequency of operation) techniques are used to uniquely distinguish received signals from different fingers. But since we are measuring distance using capacitance, fingers that are far away from each other produce signals that have very low amplitude. Having a common ground plane ensures that the calculated signals are also with respect to the same potential which means that all the fingers can be excited with the same frequency signal.

5 DESIGN OF THE CAPACITIVE PROFILING SYSTEM

The central controller board of HandSense, called CapProfiler, can be embedded inside a wrist-worn device such as a smartwatch or wristband, which leaves only electrodes on the glove. CapProfiler board follows modularized design: signal excitation, reception, as well as signal processing are all integrated on board, and the system can be put to use once the user connects glove with sensing elements with the CapProfiler board. To further lower the cost of making CapProfiler boards, we seek a design with low complexity hardware and light-weight measurement techniques in firmware.

5.1 Transmitter and receiver design

Capacitive Coupling Transmitted Signal. At any given time, HandSense transmits a sinusoidal wave as an excitation signal to an electrode and measures the received displaced current from a nearby unexcited electrode to infer the capacitive coupling between the two electrodes. The choice of transmitting frequency is dictated by several factors. On one hand, as the impedance through the air between the two electrodes is $X_C = \frac{1}{2\pi f C}$, the higher the frequency is, the lower the inter-electrode impedance is, causing more displacement current at the receiving electrode. On the other hand, higher transmitting frequency requires higher ADC sampling rates and real time processing capabilities. In HandSense, we choose 100kHz sinusoidal wave as our excitation signal.

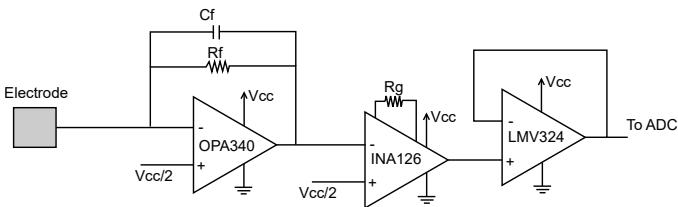


Figure 5: Analog receiver frontend

Analog receiver frontend design. We design a simple sensitive analog receiver frontend circuit connected to an electrode as

shown in Fig. 5. The displacement current measured at the receiving electrode is amplified through a transimpedance amplifier. The amplifying gain of the transimpedance amplifier (OPA340) is set by the feedback resistor R_f following the formula: $V_{out}/I_{in} = -R_f$. It also has a capacitor in parallel to create a lowpass filter to filter out unwanted higher frequency components and harmonics. Since the board is powered using a single supply a bias voltage of $V_{cc}/2$ is provided at the non-inverting terminal. This forces the DC output to about $V_{cc}/2$. The difference between this filtered, amplified output voltage and a bias voltage $V_{cc}/2$ is further amplified by a second stage using an instrumentation amplifier (INA126). This ensures that we amplify just the small received signal. The instrumentation amplifier has a default gain of 5 and additional gain can be set by using R_g . The output from the instrumentation amplifier is fed to a voltage follower with a low output impedance before being fed to the microcontroller ADC.

Multiple transmitters and receivers. We utilize a round-robin approach for multiplexing between different capacitive links, where one link is the capacitance between a pair of fingers (i.e., thumb to index finger is one link, thumb to middle finger is another link). We also observe as expected that links are symmetric (e.g., the middle-to-ring finger signal is the same as the ring-to-middle finger signal). The two multiplexers, one for transmitting and one for receiving, iteratively choose each of the electrode links, wait for the ADC to sample enough data points before switching to another link. By using the multiplexers we reuse the same signal generator and frontend receiver circuits, further simplifying our hardware design.

5.2 Estimation of received signal amplitude

A common technique to calculate the signal amplitude at a given frequency from ADC samples at a fixed sampling frequency is using Discrete Fourier Transform (DFT). However, this technique requires a high sampling rate, at least twice the frequency of interest, thus causing high processing overhead for the microcontroller. Moreover, we only need to compute signal amplitude for the transmitted frequency, thus most of the frequency spectrum produced by DFT is redundant. To avoid sampling data at high speed for high frequency signal (100KHz), we estimate the received signal amplitude by *synchronous undersampling* measurement technique, which was first proposed by Smith [37] and described in more detail in [40]. This technique can be seen as digital equivalent for synchronous detection method in analog domain.

Synchronous detection is a common measurement technique for recovering the amplitude of the received signal at the transmitted frequency. Fig. 6(a) shows a typical hardware setup for the synchronous detection measurement method. The sinusoidal signal of frequency f from the transmitting electrode induces at the nearby receiving electrode a received signal consisting of attenuated version of the transmit signal plus noise. The received signal is multiplied with the original transmitted signal to produce sidebands at $+2f$ and $-2f$ frequencies and also a DC value. A subsequent low pass filter removes these sidebands, and the remaining DC value is proportional to the amount of displacement current on the receiving electrode. This assumes the phase of received signal and transmitted signal have the same phase. In practice, the received signal is demodulated with both the transmitted signal and its 90-degree-shift version, to recover

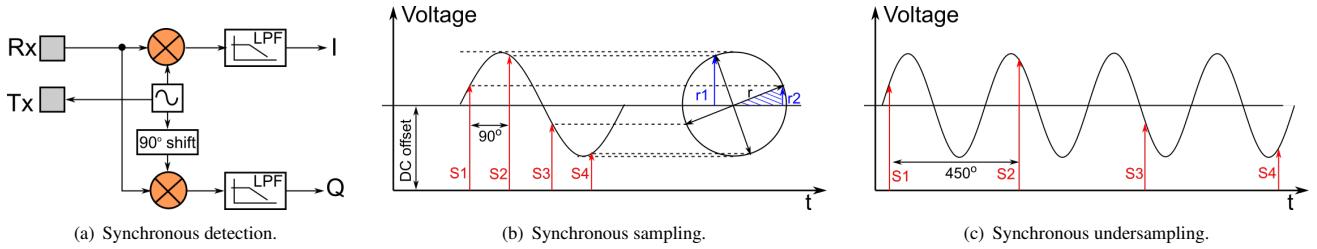


Figure 6: Measurement methods for estimation of received signal amplitude.

the in-phase (I) and quadrature (Q) components. The received signal magnitude is then calculated as $\sqrt{I^2 + Q^2}$.

Input : ADC sample array S (number of samples = 4n)

Output : Received signal amplitude

$I = Q = 0$

for $i = 0 \rightarrow n - 1$ **do**

$I = I + (S[4i] - S[4i + 2])$

$Q = Q + (S[4i + 1] - S[4i + 3])$

end

$I = I/n$

$Q = Q/n$

return $amp = \sqrt{I^2 + Q^2}/2$

Algorithm 1: Calculation of received signal amplitude using synchronous undersampling technique.

Implementing synchronous detection would require significant hardware cost (including mixers, phase shifters, and low pass filters). Moreover, the heavy low pass filtering after the mixer makes it slow to response to fast signal. *Synchronous sampling* seeks to remove these hardware components while still being able to estimate the amplitude of the received signal at the transmitted frequency.

Fig. 6(b) shows a full period of a sine wave of frequency f with DC offset. If we sample at $4f$ sampling frequency, the 4 samples on each wave cycle are separated by 90 degrees each. Let S_1, S_2, S_3, S_4 be four samples on a wave cycle, when mapping these values onto an equivalent circle, we observe that $r_1 = |S_2 - S_4|/2$ and $r_2 = |S_1 - S_3|/2$. Applying Pythagoras's law for the shaded triangle, we also have $r = \sqrt{r_1^2 + r_2^2}$. This leads to the amplitude of the sinusoidal wave can be estimated as: $r = \sqrt{(S_1 - S_3)^2 + (S_2 - S_4)^2}/2$.

The synchronous sampling technique is fast: it only requires four samples to calculate the signal amplitude. However, it requires the sampling rate of $4f$, which can exceed the capability of some microcontrollers when the transmitted frequency is high (e.g. 100KHz). To reduce the required sampling frequency, we instead use *synchronous undersampling*. We assume that inside a small time window, signal is repetitive, so instead of sampling S_1, S_2, S_3, S_4 on the same cycle, we sample them on *continuous* cycles. Now the samples are taken 450 degrees each, and the sampling frequency can be reduced to $4f/5$. The formula to estimate the signal amplitude remains the same. To increase SNR, we accumulate values of S_1, S_2, S_3, S_4 over many cycles, average them before calculating the signal amplitude. Algorithm 1 shows the full procedure.

In our implementation, we use an ADC sample array of size 16 to calculate the received signal amplitude for each link. With

sampling frequency of 80KHz, it takes 200us for capturing these 16 samples into a buffer. We implement ADC with Direct Memory Access, which frees the CPU from sampling process. In the main CPU process, we delay 1ms after switching the multiplexers to ensure the DMA buffer contains only samples after the link is stable. Therefore, a frame containing 10 measurements from 10 links takes 10ms, which leads to the measurement frequency of 100Hz in our implementation.

6 MICRO DYNAMIC FINGER GESTURE RECOGNITION

Typical finger gestures can be categorized into two groups: static gestures (such as making the victory sign, spiderman sign, okay sign) and dynamic gestures (such as swiping, sliding, tapping). HandSense focuses on the later group of gestures, especially the dynamic, micro gestures. These gestures are more suitable for interacting with the head-mounted devices for workers on manufacturing or construction sites: when the users hands can be busy with interacting with objects on the site, moving a few fingers to perform a gesture is less likely to disrupt the workflow. The gestures can be performed by finger muscles, as opposed to large hand muscle groups, thus reducing fatigue over longer use cases as well.

HandSense is highly suitable for detecting these type of dynamic, micro finger gestures. The system is capable of providing frames of link-wise measurements at rate of 100Hz, thus capturing more data points to recognize these fast, micro finger gestures. In addition, we realized that the finger movements in these gestures are more correlated with the *relative* position and velocity of each finger with regard to the other ones, as opposed to *absolute* position and velocity of individual fingers with regard to another coordinate system. Approaches using inertial sensors [8, 16, 41] or bend sensors [12, 15, 17] are able to track individual finger joints, but find it difficult to infer dynamic gestures being performed. Link-wise capacitive coupling measurements in HandSense provides better representation of these dynamic finger gestures.

In this section, we first show the three finger interactions that HandSense is able to recognize, then describe the neural network-based approaches to the problem of classifying finger gestures.

6.1 Recognition of different types of finger interactions

With the configuration of electrodes on the fingertips, we identify three finger interactions that can be recognized with HandSense. In

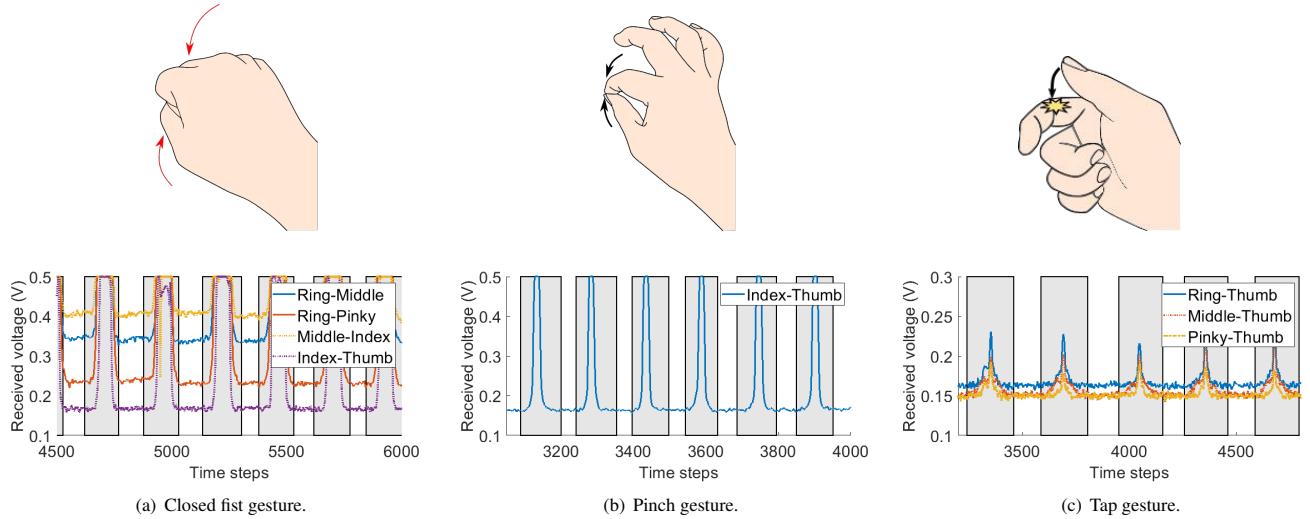


Figure 7: Illustrations of finger interactions recognized by HandSense.

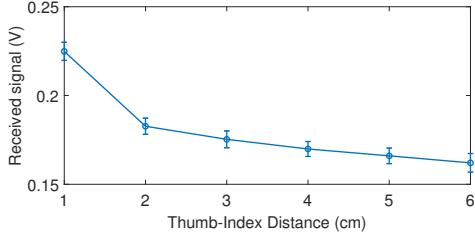


Figure 8: Received signal vs. distance (Thumb to index finger)

this section, we illustrate the signal signature of each finger interaction with an example gesture. For better visualization, we also include the time boundary for each gesture in gray boxes.

a. Direct over-the-air electrode proximity detection. Fig. 8 we show the mean and standard deviation of the received signal at a receiving electrode when the transmitting and receiving electrodes are on the index finger and the thumb respectively, and two electrodes are kept in parallel at different distances. The received signal decreases exponentially with increase in distance between transmitting and receiving electrodes. We observe that beyond 5cm, the received signal stays at a minimum level, which is the capacitive coupling between signal traces on the processing board, thus electrode distances more than 5cm are hard to detect.

As an illustration, consider the closing fist gesture shown in Fig. 7(a), in which all the fingers are curled toward the palm to make a fist. The time series of 4 links of adjacent fingers (pinky-ring, ring-middle, middle-index, index-thumb) all show signal increase as the fingers in each pair move close toward each other.

Note that since the capacitive coupling amount depends not only on distance between electrodes, but also on orientation of electrodes to each other, as well as possible capacitive coupling to the user's hand, there is no direct mapping between received signal amplitude and electrode distance. However, as we are interested in *dynamic* finger gestures, the relative change in time in each data stream is the more important feature to recognize different gestures.

b. Detection of finger touching. When the two electrodes touch each other (i.e., the insulation over the signal electrodes), the capacitive coupling between the two would be the strongest. This strong capacitive coupling produces saturated readings at the output of the frontend receiver. An example of this finger interaction is the index pinch gesture (Fig. 7(b)), in which index finger tip taps on thumb tip. The index finger electrode acts as transmitter and the thumb electrode acts as receiver. The received signal at the thumb electrode quickly increases and saturates at 0.5V when the two fingers touch each other.

c. Indirect through-the-hand electrode communication. The short range (under 5cm) of the over-the-air electrode proximity detection makes the capacitive link between far apart fingers (e.g. thumb-to-pinky) seem unusable. However, we can take advantage of the palm as a communication channel between them. We discovered that when two electrodes are touching near the center of the palm at the same time, since the human hand is conductive, there is some capacitive coupling through the hand between the two electrodes. We can take advantage of this fact to use in some intuitive and low-effort finger gestures.

As an illustration, consider the tap gesture shown in Fig. 7(c), in which the thumb taps onto the surface of the index finger. The middle, ring, and little fingers are curled into the palm and thus electrodes on these fingers are coupled to the user's hand palm region. Fig. 7(c) also shows the time series signal on three channels, from thumb to middle, ring, and little fingers, when the user performs multiple tap gestures. As can be seen in this figure, when the thumb taps the base of the index fingers, received signals on all these three channels increase because of more capacitive coupling through the hand in each link. This provides features to differentiate this gesture in the classification step.

6.2 Neural network-based gesture classification

The input to HandSense's gesture classification system is a time series data of data samples, each contains received signal amplitudes in 10 links being calculated from the CapProfiler board. There

are different approaches for the problem of time series classification [42]. Classical machine learning techniques, such as SVM, Decision Tree, Logistic Regression, require manual feature extraction from the raw sensor data before feeding into their classifiers. However, handcrafted features have several challenges, such as task or application dependence, reliance on domain knowledge, difficulty in transferring to a new type of sensor data [43].

We instead employ a data-driven approach. In particular, we seek to train end-to-end models that allow raw sensor data as input data for gesture classification. We utilize several common neural network-based methods for Time Series Classification problems, in particular Multi Layer Perceptron (MLP), Convolutional Neural Network (CNN), and Long Short-Term Memory Network (LSTM). The architectural details of each network is as follows.

Multi Layer Perceptron (MLP). As a baseline, we started with a simple neural network model as follows: Each input sequence is reshaped to a column vector of size $10 \times [\text{number of time steps}]$. The input layer is fully connected to a hidden layer, which is in turn fully connected to an output layer. The number of hidden neurons is set to approximately $2/3 \times (\text{number of input neurons} + \text{number of output neurons})$.

Convolutional Neural Network (CNN). CNN is used frequently with time series data problems, thanks to its ability to learn spatial/temporal relationship in the input data. In our experiment CNN network architecture, each input sequence is reshaped to a two-dimensional feature matrix, one dimension size is 10 (number of links being calculated), the other dimension is the time steps in the sequence. We pad input data with zeros to make input sequences of the same size. These input sequences are then used to train a Convolutional Neural Network (CNN). Our CNN consists of two convolutional layers, each followed by a max pooling layer. The kernel sizes for the convolution layers are 5×10 and 20×1 . The pool sizes are 2×1 and 20×1 . We use Rectified Linear (ReLU) activation function after each convolutional layer and dropout of rate 0.4 after the fully connected layer. The initial learning rate is set at 10^{-3} .

Long Short-Term Memory network(LSTM). LSTM is a special kind of recurrent neural network (RNN) that is capable of learning long-term dependencies. Compared to standard feedforward neural networks (e.g. MLP and CNN) that feeds the whole sequence as an entire input, LSTM is able to learn the dependencies from time-series data by feeding the sequence to the network step by step. We experimented with a LSTM network with one hidden layer consisting of 50 LSTM units. Dropout layers and L2 regularization are used to avoid over-fitting the model. We set the initial learning rate to 10^{-3} .

7 EVALUATION

In this section, we present our developed prototype, the set of dynamic, micro finger gestures used in our experiments, then describe the data collection process from users. Next, we evaluate the capability of HandSense in recognizing these gestures.

7.1 CapProfiler prototype

We designed a capacitive profiler board consisting of the following components: a Teensy 3.2 microcontroller [44] to do the central processing, a SparkFun Minigen signal generator [45], which is

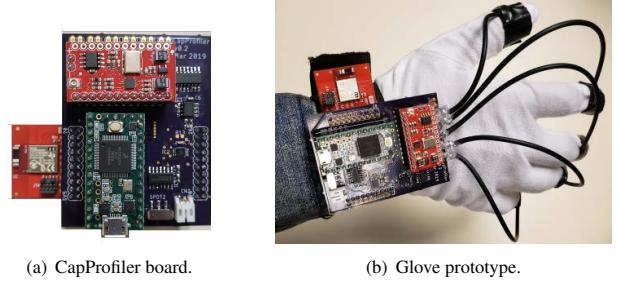


Figure 9: Prototype.

centered around the chip AD9837 [46], to generate sinusoidal wave, a custom analog receiver front-end circuit for displacement current measurements, two 8-channel CD74HC4051 multiplexers [47] for transmitting and receiving directions, and CC2650 BoosterPack [48] for Bluetooth data streaming. The signal generator generates a 1V peak-to-peak 100KHz signal. Fig. 9(a) shows the fabricated board.

We use a cotton glove and attach electrodes around its fingertips. The electrodes are connected to the central processing board by coaxial cables to avoid affect from environment noise (Fig. 9(b)).

7.2 Gesture set

HandSense is able to recognize different types of finger interactions as described above, giving us an opportunity to specify intuitive and low-effort finger gestures for operations on a head-mounted device. We design a set of such gestures, illustrated in Table 1. These gestures are highly suitable for operations on head-mounted devices, for example:

- Sliding (right to left or left to right): to rewind or fast forward any video
- Swiping: to scroll up or down a document
- Tap, double tap: to select an item on screen
- Closed fist: to close the current document or window
- Knob turn: to rotate displayed objects
- Pinch (between thumb and the remaining four fingers): to select between different options by pressing virtual buttons

The gestures in the set also demonstrate the capability of HandSense in recognizing the finger interactions described in Section 6. For example, the knob turn gestures take advantage of changing distances over time between thumb, index, and middle fingers. The pinch gestures illustrate the ability to detect the transition from far to near over-the-air electrode proximity and to the finger touching state. The through-the-hand electrode communication can be seen in gestures involving the thumb touching the palm, such as single and double touch, sliding, or swipe. Also, most gestures require only small amount of motions and can be performed by muscles controlling the fingers, rather than those involving larger muscle groups. Note that the gesture set includes certain pairs of gestures, such as sliding right to left vs. left to right, knob turn clockwise vs. counter-clockwise, which can be challenging to classify and may easily be misclassified with each other.

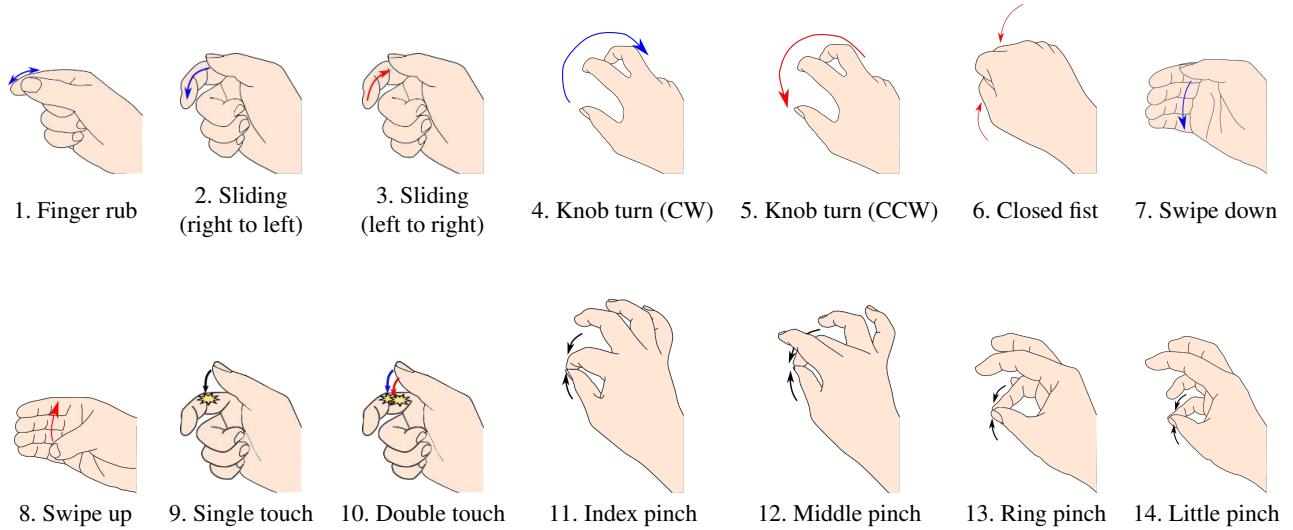


Table 1: Full gesture set used in our experiments. Note that the illustrations do not include the hand glove.

7.3 Data collection and preprocessing

Ten subjects wore the glove on the right hand and performed the gestures; the glove is equipped with electrodes on fingertips, connected with the CapProfiler board worn on the subject's wrist, as described in above section. Each gesture is captured 25 times, with experiment sessions lasting about 30 minutes per user. To simplify analysis, the start and the end of each gesture are manually marked by pressing a button. In total, we captured $10 \times 25 \times 14 = 3500$ sequences with different lengths. This dataset is used in most of our experiments.

Based on the markers of the start and the end of each gesture, sequences are extracted into individual gesture windows. The time series data is further processed by a Hampel filter, followed by a moving average filter, before being given as input to the classification system.

7.4 Gesture recognition performance

We use common metrics for multi-class classification: precision, recall and F1-score, to evaluate the performance of each model in recognizing different finger gestures. We use 10-fold cross validation for evaluating these performance metrics. Table 2 shows these metrics for three models: Multi Layer Perceptron (MLP), Convolutional Neural Network (CNN) and Long Short-Term Memory network (LSTM).

Model	Precision	Recall	F1 score
MLP	0.909	0.903	0.903
CNN	0.945	0.942	0.942
LSTM	0.976	0.975	0.975

Table 2: Classification performance of different neural network-based methods.

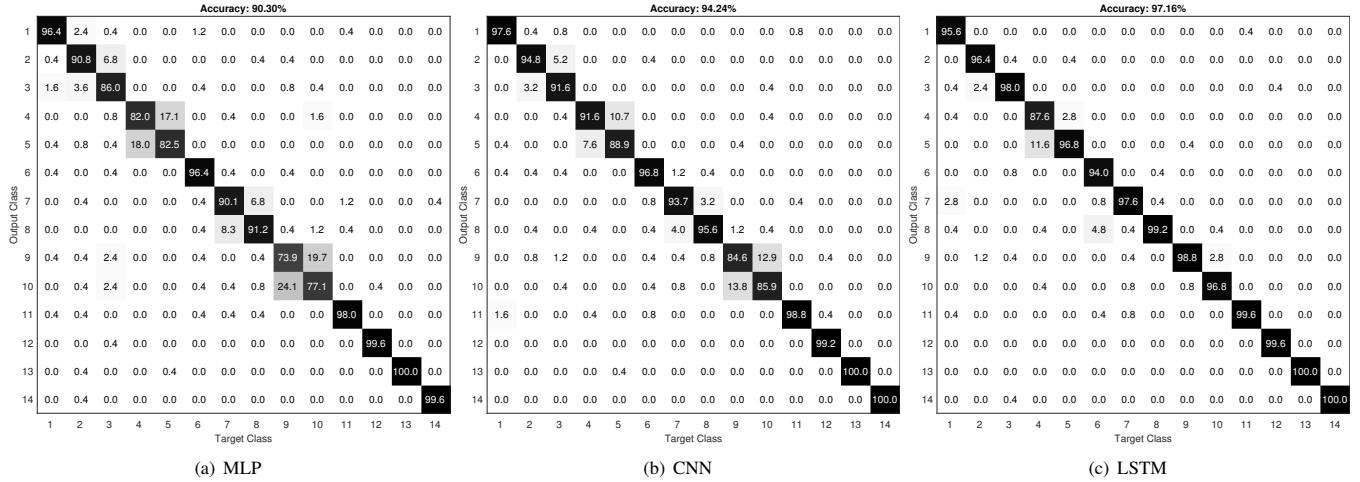
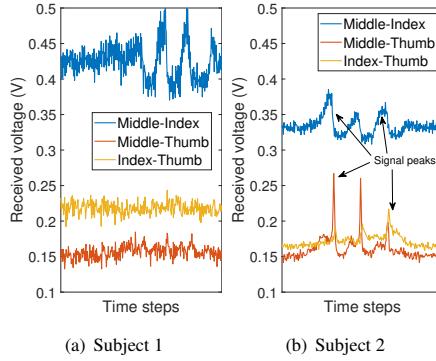
MLP achieves 0.909 precision, 0.903 recall and 0.903 F1 score, which is a good baseline for the classification. This shows that with

a simple fully connected layers model, the discriminative signatures in the capacitive profiling are already able to provide reasonably high accuracy in finger gesture classification. However, since MLP concatenates time series sequences on all the links into a single 1D input vector and samples are treated as independent neurons, it loses the temporal dependency within a single link as well as across links. For example, in swipe down gesture, the capacitive coupling amount should increase then decrease in this order: thumb-index, thumb-middle, thumb-ring, then thumb-little. As we will see, most of the misclassifications in MLP happen within close pairs of gestures: sliding right to left vs. left to right, knob turn clockwise vs. counter-clockwise, swiping up vs. down, single-touch vs. double-touch.

CNN performs better than MLP (0.945 precision, 0.942 recall and 0.943 F1 score). This is because CNN keeps the 2-dimensional data ($10 \text{ links} \times \text{number of time steps}$) as input to the network, and its 2D filters are able to learn the temporal dependency within each link (e.g. received signal rises and falls when fingers move closer and further away) as well as across links (e.g. swipe down gesture, as described above).

LSTM has been widely recognized to achieve excellent performance on time series data classification. Compared to CNN, LSTM has better memorization of the long term dependencies of the past. Based on our results in Table 2, we show that LSTM achieves the best performance in all three metrics (0.976 precision, 0.975 recall and 0.975 F1 score).

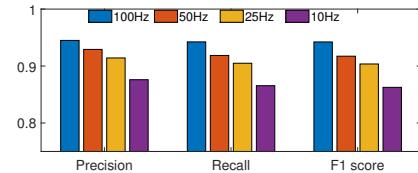
Fig. 10 shows the confusion matrix of the finger gesture classification with the three neural network-based methods, using 10-fold cross validation. As expected, we can see that most misclassifications are within pairs of close gestures: sliding left to right vs. right to left, knob turn clockwise vs. counter-clockwise. CNN performs better than MLP in differentiating gestures in each pair, thanks to its awareness of temporal dependency in the data stream. LSTM further shows its superior performance on detection of compounding gestures such as single touch and double touch. Knob turn (clockwise, counter-clockwise) are the most challenging gestures for all

**Figure 10:** Confusion matrix of finger gesture classification using three neural network-based models.**Figure 11:** Sensing data from two subjects performing knob turn (CW) gesture (gesture 4).

three models. When doing raw data analysis, we identify several possible reasons for these more likely misclassified gestures. They are hard to perform (feedback from users), meaning the collected signal might not have been consistent across users or even for the same user. Another reason is the estimation of fingertip distance in our current prototype. Fig. 11 shows two gesture instances for knob turn (clock-wise) gesture from two subjects. Intuitively, the gesture recognition for this gesture depends on distinctive features in signals in three links: middle-thumb, index-thumb and middle-index. We can see peaks in signals in middle-thumb and index-thumb links for subject 2, while these peaks disappeared in subject 1 case. This is because subject 1 has his fingers far apart when performing the gesture, while current prototype of HandSense can only recognize finger tip distance smaller than 5cm (as seen in Fig. 8). We believe the system will have better gesture recognition accuracy when longer distances can be recognized, which can be achieved by increasing the electrode size, increasing the gain and/or extending the dynamic range of the receiver. We leave this optimization to future work.

Overall, the three neural network-based methods have high gesture recognition performance on our collected dataset, proving the distinctive signatures in the data stream collected from our measurement technique. Note that this exploration of neural network-based methods is by no means an exhaustive search for the model with the highest recognition performance. Instead, the focus is on the suitability of this new measured capacitive coupling profile in recognizing fine-grained finger gestures. The results in this section provide a promising baseline, and we leave additional analysis of suitable machine learning techniques for future work.

7.5 Microbenchmarks

**Figure 12:** Effect of the measurement rate on classification performance.

Capacitive coupling measurement rate vs. classification accuracy. We evaluate the effect of the measurement rate on the classification accuracy of HandSense. From 100Hz-rate dataset collected from the above process, we downsampled the data stream to simulate data collected at 50Hz, 25Hz, and 10Hz measurement rate. On these new datasets, we use the same CNN network architecture and 10-fold cross validation to evaluate the classification performance. Fig. 12 shows Precision, Recall, and F1 scores for these measurement rates. We can see that the classification performance degrades as the measurement rate decreases. This shows the advantage of our light-weight measurement technique in delivering high-rate measurements to classify fast, dynamic finger gestures more accurately.

Glove independency. Gloves used in HandSense system serve only as a convenient means to connect finger electrodes to the Cap-Profiler board on a wrist-worn device. To illustrate that the glove

being used has little effect on the classification performance of HandSense, we asked one of the ten subjects above to wear a Hyper Tough Gripping Glove (Fig. 13) and collected another set of experiments from this subject. We then trained a CNN model using data collected from previous set of ten subjects when they wore the cotton glove, and tested this model on the newly collected data. The classification achieves 0.979 precision, 0.977 recall, and 0.977 F1 score, proving that training the HandSense classifier on only one glove allows the user to use other gloves as well.



Figure 13: Different glove.

8 DISCUSSION AND FUTURE WORK

While current prototype provides reasonably high accuracy in finger gesture recognition, future work is needed to make HandSense fit for practical applications. We identify several of such aspects in this section, including power consumption, glove usability, gesture spotting and segmentation and cross-user training.

Power consumption. For fast prototyping, our current CapProfiler prototype uses off-the-shelf modules, including a Teensy 3.2 microcontroller [44], a MiniGen signal generator module [45], and TI CC2650 BoosterPack for Bluetooth module [48]. At 3.7V supply voltage, the average current drawn in this unoptimized prototype is 90mA when Teensy is in active mode and 57mA when it is in sleep mode, with the breakdown for each component shown in Table 3. This means the CapProfiler board consumes 330mW in active mode and 211mW in sleep mode. While this is high power consumption, we believe power consumption can be reduced in an optimized prototype, given the simple functionalities of the CapProfiler board. Several power optimization methods can be: replacing MiniGen module with a simple microcontroller's pin toggle at the transmitted frequency, lowering measurement rate (increasing microcontroller's sleep time) while HandSense is in idle mode. We leave the power optimization of the CapProfiler board as the future work.

Component	Current drawn
Teensy (active mode)	38mA
Teensy (sleep mode)	5mA
Analog receiver frontend	2mA
CC2650 BoosterPack	10mA
MiniGen	40mA

Table 3: Current drawn in each component in our CapProfiler prototype.

Usability. Gloves are already prevalent in some workplace sites, such as repair and maintenance, and HandSense is easily adopted in

these areas. While the current HandSense prototype remains bulky with coaxial cables connecting the finger electrodes with the CapProfiler board, given the minimal requirements for the glove (only finger electrodes and traces are needed), we believe it is possible to design cheap gloves with all sensing elements weaved into the fabric. Also, with the advance of skin electronics [14], the electrodes and traces can be attached directly to the user's hand, thus potentially enabling more applications of HandSense in consumer electronics.

Gesture spotting and gesture segmentation. Current system assumes well-defined start and end points of each gesture as the input to the classifier. We focused more on the sensor design and the suitability of measurement signal for the task of finger gesture classification. To be able to develop HandSense into a real-world system, other challenges still remain, such as detection of registered finger gestures versus random motion, segmentation of consecutive finger gestures, which we leave for future work.

Cross user training system. The performance metrics reported in previous section is for 10-fold cross validation, which simulates a per-person trained gesture classification system. We also experimented with the leave-one-person-out approach on the same dataset, and achieved lower performance (MLP: 0.682 Precision, 0.681 Recall, 0.649 F1 Score; CNN: 0.712 Precision, 0.701 Recall, 0.671 F1 Score; LSTM: 0.822 Precision, 0.815 Recall, 0.813 F1 Score). We believe with larger dataset, a more generalized model can be built to support cross-user training scenarios.

9 CONCLUSION

In this paper, we introduce HandSense, a system based on pair-wise capacitive coupling measurements between electrodes placed on fingertips to recognize dynamic, micro finger gestures suitable for operations in Augmented Reality applications. We proposed a placement configuration for electrodes on the fingertips that minimizes the effect from the human hand to better associate the capacitive coupling measurements with inter-electrode distances. We designed a light-weight measurement technique based on synchronous undersampling to capture high-resolution capacitive profiling of fast, dynamic, micro finger gestures. The capacitive profiling is used in three end-to-end neural network-based models for gesture classification. Experiment results with our HandSense prototype show an average classification accuracy of 97% over a set of 14 dynamic, micro finger gestures from 10 different subjects. It achieves this accuracy without restrictions on hand position (as compared to cameras, for example) and with relatively lightweight instrumentation of the glove that enables use in environments where gloves are regularly changed. We believe our technique is a promising input interface to be used in conjunction with head-mounted augmented reality devices in working environments, which allows users to control the interface through finer gestures that are less interrupting to their workflow.

ACKNOWLEDGEMENTS

We thank the anonymous shepherd and the anonymous reviewers for their insightful comments. We also thank Neelakantan N. Krishnan for his help and consultation during the project. This material is based upon work supported by the National Science Foundation under Grant No CNS-1618019.

REFERENCES

- [1] Philips Azurion. <https://www.usa.philips.com/healthcare/resources/landing/azurion>.
- [2] Trimble Mixed Reality. <https://mixedreality.trimble.com/>.
- [3] A Use Case for Digital Field Service With the Microsoft HoloLens. <https://www.sikich.com/insight/microsoft-hololens-use-case-for-digital-field-service/>.
- [4] Packing with Mixed Reality: KLM uses Microsoft HoloLens to redefine its cargo training experience with mixed reality. <https://customers.microsoft.com/en-gb/story/klm-airlines-travel-and-transportation-hololens-azure-netherlands>.
- [5] Microsoft HoloLens 2. <https://www.microsoft.com/en-us/hololens>.
- [6] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 851–860, New York, NY, USA, 2016. ACM.
- [7] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyammath Gollakota. Fingerio: Using active sonar for fine-grained finger tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1515–1525, New York, NY, USA, 2016. ACM.
- [8] Hongyi Wen, Julian Ramos Rojas, and Anind K. Dey. Serendipity: Finger gesture recognition using an off-the-shelf smartwatch. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3847–3851, New York, NY, USA, 2016. ACM.
- [9] Yang Zhang and Chris Harrison. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, UIST '15, pages 167–173, New York, NY, USA, 2015. ACM.
- [10] Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. Finexus: Tracking precise motions of multiple fingertips using magnetic sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 1504–1514, New York, NY, USA, 2016. ACM.
- [11] Tianxing Li, Xi Xiong, Yifei Xie, George Hito, Xing-Dong Yang, and Xia Zhou. Reconstructing hand poses using visible light. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(3):71:1–71:20, September 2017.
- [12] Thomas G. Zimmerman, Jaron Lanier, Chuck Blanchard, Steve Bryson, and Young Harvill. A hand gesture interface device. In *Proceedings of the SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface*, CHI '87, pages 189–192, New York, NY, USA, 1987. ACM.
- [13] Laura Dipietro, Angelo M Sabatini, and Paolo Dario. A survey of glove-based systems and their applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(4):461–482, 2008.
- [14] The Future of Skin Electronics. <https://www.youtube.com/watch?v=zpGujcLRHNw>.
- [15] Gary Grimes. Digital data entry glove interface device. <https://patents.google.com/patent/US4414537A/en>, (1981).
- [16] Jose L. Hernandez-Rebolledo, Nicholas Kyriakopoulos, and Robert W. Lindeman. The acceleglove: A whole-hand input device for virtual reality. In *ACM SIGGRAPH 2002 Conference Abstracts and Applications*, SIGGRAPH '02, pages 259–259, New York, NY, USA, 2002. ACM.
- [17] Cyberglove ii. <http://www.cyberglovesystems.com/cyberglove-ii>.
- [18] Tushar Chouhan, Ankit Panse, Anvesh Kumar Voonia, and SM Sameer. Smart glove with gesture recognition ability for the hearing and speech impaired. In *2014 IEEE Global Humanitarian Technology Conference-South Asia Satellite (GHTC-SAS)*, pages 105–110. IEEE, 2014.
- [19] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. utrack: 3d input using two magnetic sensors. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 237–244. ACM, 2013.
- [20] Hoang Truong, Shuo Zhang, Ufuk Muncuk, Phu Nguyen, Nam Bui, Anh Nguyen, Qin Lv, Kaushik Chowdhury, Thang Dinh, and Tam Vu. Capband: Battery-free successive capacitance sensing wristband for hand gesture recognition. In *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, pages 54–67. ACM, 2018.
- [21] Jun Rekimoto. Gesturewrist and gesturepad: Unobtrusive wearable interaction devices. In *Proceedings Fifth International Symposium on Wearable Computers*, pages 21–27. IEEE, 2001.
- [22] Yang Zhang, Robert Xiao, and Chris Harrison. Advancing hand gesture recognition with high resolution electrical impedance tomography. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 843–850, New York, NY, USA, 2016. ACM.
- [23] Yang Zhang and Chris Harrison. Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology*, pages 167–173. ACM, 2015.
- [24] Munehiko Sato, Ivan Poupyrev, and Chris Harrison. Touché: enhancing touch interaction on humans, screens, liquids, and everyday objects. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 483–492. ACM, 2012.
- [25] Hrvoje Benko and Andrew D Wilson. Depthtouch: using depthsensing camera to enable freehand interactions on and above the interactive surface. In *In Proceedings of the IEEE Workshop on Tabletops and Interactive Surfaces*. Citeseer, 2009.
- [26] Robert Wang, Sylvain Paris, and Jovan Popović. 6d hands: markerless hand-tracking for computer aided design. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 549–558. ACM, 2011.
- [27] Cem Keskin, Furkan Kiraç, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *European Conference on Computer Vision*, pages 852–863. Springer, 2012.
- [28] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.
- [29] Cheng Zhang, Qiyue Xue, Anandghan Waghmare, Sumeet Jain, Yiming Pu, Sinan Hersek, Kent Lyons, Kenneth A. Cunefare, Omer T. Inan, and Gregory D. Abowd. Soundtrak: Continuous 3d tracking of a finger using active acoustics. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 1(2):30:1–30:25, June 2017.
- [30] Tianxing Li, Xi Xiong, Yifei Xie, George Hito, Xing-Dong Yang, and Xia Zhou. Reconstructing hand poses using visible light. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):71, 2017.
- [31] Jon Moeller and Andruid Kerne. Zerotouch: an optical multi-touch and free-air interaction architecture. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2165–2174. ACM, 2012.
- [32] Jess McIntosh, Asier Marzo, and Mike Fraser. Sensir: Detecting hand gestures with a wearable bracelet using infrared transmission and reflection. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology*, UIST '17, pages 593–597, New York, NY, USA, 2017. ACM.
- [33] Wenfeng He, Kaishun Wu, Yongpan Zou, and Zhong Ming. Wig: Wifi-based gesture recognition system. In *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7. IEEE, 2015.
- [34] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. Wigest: A ubiquitous wifi-based gesture recognition system. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1472–1480. IEEE, 2015.
- [35] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. Signfi: Sign language recognition using wifi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1):23:1–23:21, March 2018.
- [36] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu-Han Kim. Widraw: Enabling hands-free drawing in the air on commodity wifi devices. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 77–89. ACM, 2015.
- [37] Joshua Smith. *Electric Field Imaging*. PhD thesis, Massachusetts Institute of Technology, 1999.
- [38] Worker deaths by electrocution. <https://www.cdc.gov/niosh/docs/98-131/pdfs/98-131.pdf>.
- [39] Shubho Banerjee and Mason Levy. Approximate capacitance expressions for two equal sized conducting spheres. *Proc. ESA Annu. Meet. Electrostatics*, 2014.
- [40] Synchronous sampling and algorithmic amplitude detection. https://www.eetimes.com/document.asp?doc_id=1253691#.
- [41] Henk G Kortier, Victor I Sluiter, Daniel Roetenberg, and Peter H Veltink. Assessment of hand kinematics using inertial and magnetic sensors. *Journal of neuroengineering and rehabilitation*, 11(1):70, 2014.
- [42] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, Mar 2019.
- [43] Henry Nweke, Teh Wah, Mohammed Ali Al-garadi, and Uzoma Alo. Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges. *Expert Systems with Applications*, 105, 04 2018.
- [44] Teensy 3.2. <https://www.pjrc.com/store/teensy32.html>.
- [45] SparkFun MiniGen. <https://www.sparkfun.com/products/11420>.
- [46] Analog devices ad9837 datasheet. <https://www.analog.com/media/en/technical-documentation/data-sheets/AD9837.PDF>.
- [47] Texas instruments cd74hc4051 datasheet. <http://www.ti.com/lit/ds/symlink/cd74hc4051-ep.pdf>.
- [48] TI SimpleLink™ Bluetooth® low energy CC2650 Module BoosterPack™ Plug-in Module. <http://www.ti.com/tool/BOOSTXL-CC2650MA>.