Automatic Calibration of Magnetic Tracking

Mingke Wang[◇*], Qing Luo[†*], Yasha Iravantchi[◇], Xiaomeng Chen[†]
Alanson Sample[◇], Kang G. Shin[◇], Xiaohua Tian[†], Xinbing Wang[†], Dongyao Chen[†]

[†] Shanghai Jiao Tong University
{luoqing,sjtu_chenxm,xtian,xwang8,chendy}@sjtu.edu.cn
[◇] University of Michigan, Ann Arbor
{mingkew,yiravan,apsample,kgshin}@umich.edu

ABSTRACT

Magnetic sensing is emerging as an enabling technology for various engaging applications. Representative use cases include high-accuracy posture tracking, human-machine interaction, and haptic sensing. This technology uses multiple MEMS magnetometers to capture the changing magnetic field at a close distance. However, magnetometers are susceptible to real-world disturbances, such as hard- and soft-iron effects. As a result, users need to perform a cumbersome and lengthy calibration process frequently, severely limiting the usability of magnetic tracking.

To remove/mitigate this limitation, we propose MAGIC (**MAG**netometer automat**Ic C**alibration), a systematic framework to automatically calibrate both soft- and hard-iron disturbances for a MEMS magnetometer array. To minimize the need for user intervention, we introduce a novel auto-triggering module. Unlike the legacy manual calibration method, MAGIC achieves superior calibration performance (e.g., for tracking applications) with minimal user attention. Via empirical studies, we show MAGIC also incurs marginal overhead and cost, such as a total energy cost of 0.108 J.

CCS CONCEPTS

• **Hardware → Sensor devices and platforms**.

KEYWORDS

MEMS magnetometer; calibration

1 INTRODUCTION

Magnetic tracking of passive magnets as battery-free "markers" is a recent, emerging way to support novel interactive applications. Compared to other sensing modalities, such as computer vision,

---

*Mingke Wang and Qing Luo are co-primary authors. Mingke's engagement with this project started when he was an undergraduate student at SJTU .

RFID, and motion sensors, magnetic tracking maintains robust tracking across various environments and situations. For example, magnet-based hand gesture tracking [13–15, 43] can reliably track hand/finger movements in non-line-of-sight (NLOS) situations. Furthermore, magnetic tracking offers a fine-grained resolution to closely monitoring knee movements [34], subtle changes of muscle length [44], and can even sense tactile events such as touches through a magnetic "skin" [10].

Magnetic tracking often uses a MEMS magnetometer [27] array to find the 3D position of nearby passive magnet(s). For example, [13] used an eight-sensor, wearable and untethered array. [14] proposed a two-sensor, finger-worn array. Compared to other magnetic sensors, such as fluxgate sensor [36], MEMS magnetometers are compact, energy-efficient and low-cost which is essential for real-world apps.

However, MEMS magnetometers require frequent manual calibration, undermining the ability to track magnets over time. As seen in Sec. 2, commodity MEMS magnetometers suffer from continuous, intrinsic drift. These erroneous sensor readings, if left unaddressed, increasingly degrade tracking performance and applications.

The root causes of this problem are hard- and soft-iron effects [30] in MEMS magnetometers. Specifically, the hard-iron effect is caused by magnetic objects with a constant additive magnetic field (e.g., household magnets). The soft-iron effect is due to ferromagnetic objects that distort (e.g., iron and nickel) an existing magnetic field. Thus, hard- and soft-iron effects cause data offset and scale distortion, respectively, in magnetometer sensor readings. To overcome hard- and soft-iron disturbances, a thorough and accurate calibration is necessary for magnetic tracking and localization applications.

Unfortunately, state-of-the-art calibration procedures are cumbersome as users have to manually hold the sensing platform and make lengthy movements (such as a figure-8 motion) — a necessary step to have diverse readings from all three axes (i.e., x, y, z) of the sensor. Note that the existing calibration methods are predominantly designed for handheld devices, thus making them infeasible for magnetic tracking apps. For example, a sensing array may be worn on the lower body that cannot be easily dismantled or moved to perform motion-based calibration. Thus, the calibration procedure can severely undermines the usability of magnetic tracking in practice.

To achieve a reliable and usable calibration performance, we start with answering the following two questions.
Q1: Can we calibrate the magnetometer array with minimal user involvement? As we will elaborate in Sec. 3, the movement-based calibration severely undermines the usability of magnetic
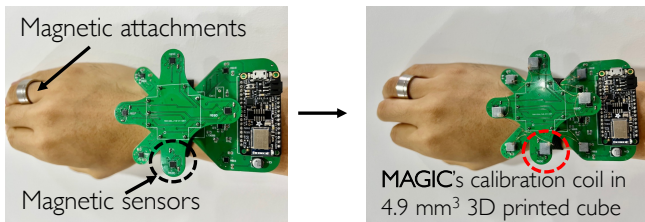
Figure 1: Applying MAGIC on magnetic tracking systems only incurs minimal overhead.

tracking apps. A practical solution should enable quick calibration of soft- and hard-iron effects with minor user involvement.

Q2: Can we trigger the calibration procedure automatically? A practical magnetic sensing system should be able to detect whether or not it needs to correct soft/hard-iron errors. This feature has two advantages: it (1) enables calibration without interrupting its regular usage (i.e., without alerting and requiring the user to manually calibrate); (2) improves the efficiency of the calibration procedure. These are essential as mobile magnetic tracking apps demand low latency, high sensing accuracy, and low power consumption.

In this paper, we present MAGIC, a method for automatically calibrating MEMS magnetometers with minimal user involvement, power consumption, and computational overhead. MAGIC can provide users a calibrate-as-you-use (CAYU) experience, thus achieving high usability and reliability of magnetic sensing. MAGIC is comprised of two key modules:

- **Calibration Module:** calibrates both hard- and soft-iron disturbances with minimal user intervention.
- **Auto-triggering Module:** automatically determines if the sensing platform needs calibration.

We eliminate/mitigate the need for manual calibration by introducing a novel calibration algorithm for high accuracy magnetic tracking platforms using multiple magnetometers. MAGIC can fully calibrate the soft- and hard-iron effects without any movement of the platform and is resilient to the *soft-iron anomaly* (as we will articulate in Sec. 3.2), a soft-iron mis-prediction that can severely undermine the performance of magnetic sensing. To the best of our knowledge, MAGIC is the first practical solution to address the soft-iron anomaly.

For the auto-triggering module (Sec. 5), we propose a lightweight algorithm to efficiently detect environmental changes *without* interrupting the regular usage of the sensing platform. The form factor of MAGIC's calibration module can also be integrated with existing magnetic sensing platforms. As illustrated in Fig. 1, MAGIC can be easily installed on existing magnetic tracking systems [13]. We design hardware based on its principles; specifically, we build a solenoid magnetic coil for each magnetometer to generate the calibrated magnetic field. We stabilize the coil with a 3D-printed stand and designed a current-controlled driving circuit for the coils, as detailed in Sec. 6.

Sec. 7 evaluates MAGIC's reliability and usability via empirical studies. Specifically, our experimental results show MAGIC to be able to fully calibrate both hard- and soft-iron effects. For analytical, filter- and machine-learning-based tracking schemes, MAGIC's calibration outperforms the lengthy full-sphere calibration with less overhead (e.g., less user's movement and attention).

This paper makes the following contributions. (1) Development of a novel calibration pipeline, MAGIC, for handling both hard- and soft-iron disturbances on mobile magnetic sensing systems; (2) proposal of an auto-triggering scheme for determining when to calibrate; and (3) evaluation of MAGIC's calibration performance with a functioning prototype.

2 BACKGROUND

We first cover the background of hard- and soft-iron effects, and elaborate on the differences between the new emerging magnetic sensing and the e-compass applications. Next, we articulate how COTS devices tackle the calibration problem. Finally, we analyze the calibration issue for existing magnetic sensing techniques.

2.1 Why is Calibration Needed?

A MEMS magnetometer suffers from disturbances from both external magnetic environments (hard- and soft-iron effects) and internal drift. A combination of these two severely distorts and alters the sensor readings, resulting in poor tracking performance.

2.1.1 Hard-iron effect. The hard-iron effect [9] is generated when permanently magnetized ferromagnetic objects [30] create an induced magnetic field in surrounding ferromagnetic materials. In magnetometers, the generated magnetic field causes a permanent bias, adding a *constant offset* to the sensor readings.

2.1.2 Soft-iron effect. The soft-iron effect [37] stems from the induced magnetic field by un-magnetized ferromagnetic materials, such as iron and steel. These ferromagnetic components [30] will generate an induced magnetic field proportionally to the local magnetic field, and thus distort the magnetometer readings. The hard-iron effect is usually modeled as a 3D offset of the magnetic field, while the soft-iron effect is strongly associated with the axis and usually modeled as a six-component symmetric matrix.

Besides the soft- and hard-iron effects incurred in an outdoor magnetic environment, the magnetometer itself also suffers from an internal drifting problem. We conducted a simple experiment by placing a small magnet near an LSM9DS1 MEMS magnetometer [2]. The magnetometer and its environment are kept intact during the data collection, as shown in Fig. 2. Our result indicates that the magnetometer reading gradually changes in both settings only within five minutes. Note that the observed drift is not due to the change of earth magnetic field, which only has a less than 25 nanotesla (nT) daily variation [41]. Therefore, if the magnetometer is used without calibration, the hard-/soft-iron effect will yield erroneous data that may cause systems like e-compass [35] to fail.

2.2 Demystifying Existing Magnetometer Calibration Methods

We start with answering the following key questions.

Q1: How do COTS devices calibrate their magnetometers? When calibration is needed, the system will instruct the user to perform a "figure-8" movement [3] repetitively. This movement is part of a magnetometer calibration routine using the Kasa sphere fitting algorithm [23] which efficiently computes a sphere's center and radius by solving a set of linear equations under the assumption that the magnetometer's reading is only from the earth's magnetic
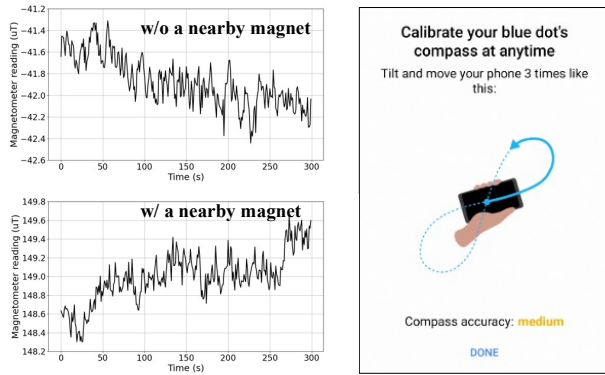
Figure 2: Drifts of magnetometer, w/ and w/o a nearby magnet, we present the x-axis data. An example of magnetometer drift is smartphone compass drift.

field. By instructing the user to repetitively make a figure-8 movement, the magnetometers take readings at different angles for the algorithm to perform a more accurate spherical fitting. This calibration is required often and implemented within the Android source code [6]

Q2: Aren't existing COTS devices (e.g., smartphone and UAVs) already telling users when to calibrate? Existing electronic devices, such as smartphones, can prompt pop-up messages to warn users of poor magnetometer accuracy. For example, COTS mobile devices usually notify users with a message "the device's compass needs calibration".

We examine the COTS method by analyzing the Android calibration code base [6]. Specifically, there is no built-in detection system to automatically determine whether the soft/hard-iron error should be handled via calibration. Instead, a global flag `calibration_quality` is used to indicate if there is any update of the magnetometer's parameter since the last system reboot. The default value for "calibration_quality" is `UNDETERMINED`, suggesting there is no magnetometer calibration, and hence may yield inaccurate sensor readings. The calibration is implemented by calling an "online calibrate" function, which derives the calibration parameter with the user's figure-8 shape movement and Kasa fitting algorithm. After the Kasa fitting, "calibration_quality" will be set to `HIGH_QUALITY`, indicating the magnetometer reading has been calibrated.

2.3 Premier of Magnetic Tracking

Magnetic tracking is an emerging sensing technology that utilizes the near-field magnetic field for high-accuracy tracking [13, 43], positioning [34], or pattern recognition [10]. Compared to state-of-the-art tracking approaches, magnetic tracking achieves high accuracy at low cost. Specifically, computer vision approaches [24, 38] fall short due to their occlusion (i.e., NLOS) problem. RF-based approaches [28, 46] may suffer from the multipath problem, thus undermining their performance of robust tracking. Inertial measurement units (IMUs) can achieve embedded motion tracking [39, 40], but suffer from the drift problem, incurring an accumulated error. Existing magnetic tracking systems usually use multiple COTS magnetometers to track the varying magnetic field, which is induced by passive magnet, electromagnet, or ferromagnetic objects.
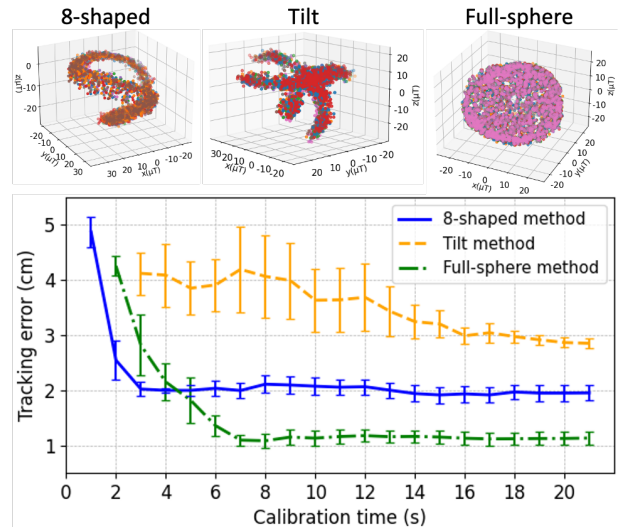

Figure 3: Comparison of existing calibration methods.

Compared to other tracking and sensing techniques, magnetic tracking has the following challenges/features:

(1) Magnetic tracking needs to carefully address the noisy sensor data. Specifically, magnetic field strength decreases rapidly (i.e., inverse cube [31]) with the distance between the magnet and the sensor array.
(2) Sensitive to soft- and hard-iron changes. That is, a subtle change of soft- and hard-iron parameters can drastically influence the sensing accuracy. We will further elaborate their impact in Sec. 7.1.
(3) The sensory platform is usually built with an array of MEMS magnetometers.
(4) The background magnetic field is usually treated as a global offset to the sensor readings.

Note that these features differentiate magnetic tracking apps from conventional e-compass apps. Next, we articulate magnetometer calibration methods and why legacy approaches are ill-suited for magnetic sensing apps. In MAGIC, we mainly focus on solving the calibration issue for passive magnet tracking [13, 43].

The passive magnet tracking algorithm works by sensing the varying spatial distribution of magnetic field modeled by dipole model [12]. For example, [29] proposed a lightweight finger tracking system by attaching magnet on the user's finger tip, and MagX [13] is a high accuracy mobile magnetic system for hand tracking. We use the open source MagX project as our testbed to validate MAGIC's performance. MagX separates the magnetic field generated by the passive magnet from the background magnetic field by using Levenberg-Marquardt (LM) algorithm. The magnet field observed on the sensor can be represented as a linear combination of the magnetic field generated by each magnet and the background magnetic field as expressed in Eq. (1). The problem is to estimate the magnet position and orientation given the sensor reading. We refer interested readers to [32] for more details.

3 WHY MAGIC?

3.1 Performance of Existing Methods

We conducted an empirical study of the performance of existing calibration methods on a magnetic tracking platform, using the public repository of [13]. We employed the ellipsoid-fitting method for calibration and three representative calibration movements, i.e., 8-shaped, tilting, and full-sphere methods. Note Google recommends 8-shaped and tilting methods for smartphones and tablets [3]. In the full-sphere movement, users need to evenly distribute magnetic sensor readings on the surface of a sphere. For this, the user has to rotate the sensor array thoroughly. Fig. 3 shows the distribution of sensor readings of three movements.

For further investigation of the usability of these methods, we recruited three volunteers of age between 21 to 30. The average amounts of time taken for 8-shaped, tilting, and full-sphere calibrations were approximately 2.3, 4, and 5 seconds. Next, each volunteer calibrates the sensor using the three different methods. Finally, we apply the derived hard- and soft-iron parameters for each calibration process for the magnet tracking app. Specifically, we collect the magnetometer readings and ground truth with MagX and Leap Motion (as indicated in [13]), respectively.

Fig. 3 shows the tracking accuracy of 21 cm. Specifically, all calibration methods can mitigate the tracking error rate. The 8-shaped movement converges faster with a better tracking performance than the tilting method. The full-sphere method shows the best tracking accuracy despite its longest convergence time. Hence, the existing calibration methods require user involvements to achieve high sensing accuracy, thus limiting their usability and reliability.

3.2 Limitations of State-of-the-Arts

Navigation apps using the e-compass are the predominant usage scenario for conventional calibration methods on mobile devices. The calibration only needs to correct the magnetic field's direction (e.g., North or South poles). In practice, the calibration process projects sensor readings to a *unit* sphere.

Compared to the e-compass app, accurate information on the radius of sphere is essential for magnetic sensing apps. Therefore, the radius should reflect the earth's magnetic field rather than just a unit scalar. For example, the authors of MagX [13] conducted an 8-shaped motion calibration and then used sphere fitting for each magnetometer. The earth magnetic field G is estimated as the mean of the radii of all fitted spheres $G = \frac{\sum_{i=1}^{N} r_i}{N}$. The soft-iron parameter for magnetometer i is derived as $\alpha_i = \frac{r_i}{G}$, and the calibrated sensor reading is $\hat{B} = (B - \beta)/\alpha$, where β is the hard-iron parameter.

However, it falls short if there is a nontrivial soft-iron distortion. Let us consider two magnetometers with different soft-iron distortions. Suppose the actual G is $50\mu T$, then the estimation $\hat{G} = \frac{1}{2}(r_1 + r_2) = \frac{1}{2}(\frac{1}{2}G + \frac{1}{1}G) = \frac{3}{4}G$ deviates $\approx 25\%$ from the ground truth. We call this estimation error *soft-iron anomaly*. Thus, existing calibration methods are ill-suited for multi-magnetometer sensing systems.

3.3 Overview of MAGIC

MAGIC is a self-calibration scheme that can correct both soft- and hard-iron disturbances with minimal user involvement. As shown
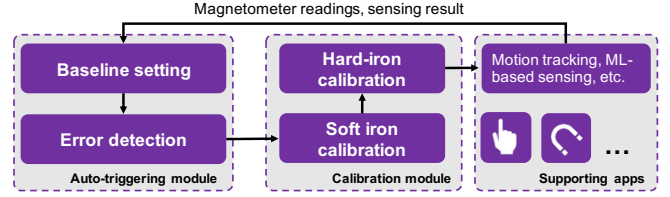

Figure 4: Overview of integrating MAGIC with tracking apps.

in Fig. 4, MAGIC consists of two key modules: (a) the calibration module for correcting both hard- and soft-iron disturbances and (b) the auto-triggering module to detect if calibration is needed. In Secs. 4 and 5, we provide the theoretical underpinnings of MAGIC's calibration and auto-triggering, respectively.

4 SOFT AND HARD-IRON CALIBRATION

We first articulate the theoretical underpinning of soft-iron calibration and then present a novel hardware–software co-design of soft-iron calibration. Finally, we introduce our hard-iron calibration method.

4.1 Soft-iron Calibration

For calibration against soft-iron disturbances, the key idea is to use a controllable magnetic field to achieve deterministic readings on magnetometers. By exploring the relationship between the sensor reading and the deterministic magnetic field, we can derive the soft-iron parameters.

A controllable magnetic field can be obtained with electromagnet(s). Suppose there is one electromagnet near the platform, then the reading of magnetometer i can be:

$$B_i = G + \frac{\mu_0}{4\pi}\left(\frac{3(m \cdot r_i)r_i}{|r_i|^5} - \frac{m}{|r_i|^3}\right) \qquad (1)$$

where the G is the background magnetic field, and $\vec{r}$ is the distance between the sensor and the electromagnet.

If the relative position and orientation between the electromagnet and the sensors do not change, the magnet moment should be proportional to the current. Now, the right-hand side is a function of the current:

$$B_i = G + h(I). \qquad (2)$$

Since the magnetometer experiences both soft- and hard-iron disturbances, its readings can be modeled as:

$$\hat{B}_i = \alpha B_i + \beta, \qquad (3)$$

where α (β) is the soft(hard)-iron effect, B_i is the ideal sensor reading, and $\hat{B}_i$ is the observed sensor reading. Eq. (3) now is:

$$B_i = \gamma \hat{B}_i + \lambda \qquad (4)$$

Assume G remains constant during the calibration process, with Eq. (2), one can then convert Eq. (4) to:

$$\gamma \hat{B} + \omega = h(I), \qquad (5)$$

where $\omega = \lambda - G$. By applying different current I, one can have a system of linear equations. The parameters γ and ω can be solved

efficiently. Unlike other calibration methods, MAGIC's soft-iron calibration algorithm calculates the soft-iron parameter directly, making it *insusceptible* to *soft-iron anomaly* (as we elaborated in Sec. 3.2).

4.2 Hard-iron Calibration

After fixing the soft-iron distortion, MAGIC corrects the hard-iron offset. First, we express the reading of sensor i as a linear combination of the signal of interest (B^{signal}), the background magnetic field (G), and the hard-iron offset (β_i):

$$\hat{B}_i = G + \beta_i + B^{signal}. \tag{6}$$

Since β_i is a constant component, we employ an efficient "reset" process to eliminate the hard-iron drift. Specifically, if B^{signal} is negligible (e.g., $B^{signal} \approx 0$) during the calibration, the sensor reading can be expressed as:

$$\hat{B}_i^{cali} = G_0 + \beta_i, \tag{7}$$

where G_0 is the background magnetic field during calibration.

After calibration, the sensor reading can be:

$$\hat{B}_i' = \hat{B}_i - \hat{B}_i^{cali} = B^{signal} + G - G_0 = B^{signal} + G', \tag{8}$$

where G' is the shifted background magnetic field.

In the passive magnet tracking system with the LM algorithm, B^{signal} is the magnetic field generated by the passive magnet. The cost function of the LM algorithm after hard-iron calibration is:

$$E_i = B_i - (\hat{B}_i - \hat{B}_i^{cali}), \tag{9}$$

where

$$\hat{B}_i - \hat{B}_i^{cali} = \frac{\mu_0}{4\pi}\left(\frac{3(m \cdot r_i)r_i}{|r_i|^5} - \frac{m}{|r_i|^3}\right) + G - G_0 \tag{10}$$

which only includes the magnetic field generated by the magnet and the shifted earth magnetic field, thus effectively alleviating the hard-iron offset.

5 AUTOMATIC TRIGGERING

As a major application of magnetic sensing, activity tracking [13, 43], requires continuous and in situ usage experience. To meet this requirement, we propose an auto-triggering of calibration. As shown in Fig. 4, this automatic triggering consists of two components: baseline setting and detection. The baseline setting derives a novel set of features to characterize the magnetic state of a sensory platform. Based on the thus-derived baseline, the second component uses an efficient decision tree to determine whether a soft/hard-iron calibration is needed.

5.1 Baseline Setting

MAGIC enters the baseline setting stage after configuring soft- and hard-iron parameters. The baseline setting stage utilizes the magnet tracking results and the background magnetic field, i.e., G.

We propose use of the following features to characterize the sensing platform:

- (α, β): α and β are the scale and the offset derived from the sphere-fitting algorithm by using G.
- MSE_{base}: $\sum_i [(B_i - \beta)^2 - \alpha^2]$: the mean square error (MSE) of the fitting process.

To ensure the baseline setting stage has sufficient data for sphere fitting, we adopt the diversity checker mechanism [7] used in Android OS. The algorithm calculates the minimum distance d from G_t (i.e., G at time t) to all collected G.

$$d_t = \min_{G_i \in \mathbb{G}} ||G_t - G_i||_2, \tag{11}$$

where $\mathbb{G}$ is the set of collected G. G_t will be added to the collected data only when d_t is larger than the distance threshold $\mathcal{T}_{dis}$. We invoke the sphere fitting algorithm once the buffer is full, i.e., the amount of accumulated data reaches $\mathcal{T}_{buffer}$. Note that $\mathcal{T}_{dis}$ and $\mathcal{T}_{buffer}$ are hyperparameters that only need one-time factory reset.

We will elaborate the baseline setting stage with the empirical study in Sec. 5.2. After tuning suitable parameter(s) for $\mathcal{T}_{dis}$ and $\mathcal{T}_{buffer}$, one can efficiently conduct the baseline setting with a marginal time overhead.

5.2 Detection Stage

After obtaining the baseline, the detection stage will be triggered *periodically* to check if a re-calibration is needed. The sensor readings and the tracking results at each timestamp will be stored and grouped into *frames* for the detection stage. The detection algorithm will be applied to each frame and output a binary decision, indicating whether the sensing platform needs to be re-calibrated or not.

If the soft/hard-iron parameter changes, then the sensor reading will be distorted, thus increasing the tracking error of the magnet position/orientation and the background magnetic field (G). Specifically, we have made two observations from real-world soft-/hard-iron disturbances. First, there is a large variation (e.g., a zigzag pattern) in the tracking trajectory of the magnet. Second, the derived G will be distorted compared to that obtained from the baseline setting stage. For example, the derived sphere would be shifted or distorted if the soft-/hard-iron environment changes (as shown in Fig .5(a)), thus affecting the derived offset and scale.

We therefore design the following set of features for reflecting the varying soft-/hard-iron effects on the sensing platform.

- $avg\left(|P_{mag}|\right)$: average Euclidean distance from the magnet to the original point;
- $(\hat{\alpha}, \hat{\beta})$: the sphere-fitting result by using the background magnetic field;
- $(|\hat{\alpha} - \alpha|, |\hat{\beta} - \beta|)$: the difference in scale and offset; and
- MSE_{detect}: $\sum_i [(\hat{G}_i - \beta)^2 - \alpha^2]$.

We use $avg\left(|P_{mag}|\right)$ for determining the reliability of magnet tracking results. $\hat{\alpha}$, $\hat{\beta}$, MSE, and $|\hat{\alpha} - \alpha|$, $|\hat{\beta} - \beta|$ depict the changes of the background magnetic field. If the MSE is larger than the baseline, we can conclude that the soft/hard-iron parameters are changed (as shown in Fig .5(b)).

The detection stage leverages these features to determine if the platform needs calibration. Specifically, we only consider the system running normally when (a) $|P_{mag}| <= 40$; (b) $|\hat{\alpha} - \alpha| < k_1$, $|\hat{\beta} - \beta| < k_2$; (c) $MSE_{detect} < k_3 MSE_{base}$.

According to our empirical study (Sec. 7.3), by tuning these thresholds (i.e., k_1, k_2, and k_3) and using a grid search, MAGIC can achieve 75% F-1 score with almost no power consumption. Note that

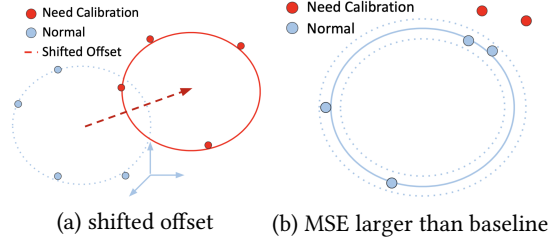(a) shifted offset   (b) MSE larger than baseline

Figure 5: Detecting the change of soft/hard-iron effects.

these hyperparameters are *constant* for apps. Tuning the parameters is a one-time effort (e.g., factory setting) for the users.

6 HARDWARE IMPLEMENTATION

To facilitate the thus-proposed calibration scheme in the real world, MAGIC's hardware must meet two design goals. First, MAGIC should enable a controllable and evenly distributed calibrating signal on three axes of each sensor. Second, the calibration should incur low overheads, especially in energy consumption and installation cost.

We employ magnetic coils to achieve the first goal. According to Ampére's law in electromagnetism [17], one can control the coil's magnetic field (B^{coil}) by tuning the current, the number of wraps, and coil length. Specifically, $B^{coil} = \mu_0 nI$, where μ_0 is the magnetic constant, n is the number of wraps of the coil, and I is the current. We design a magnetic coil for *each* sensor in MAGIC. Next, we elaborate on three key design elements: coil, form factor, and calibrating signal.

6.1 Coil Design

We apply air-core solenoid coils for generating a magnetic field. To generate an evenly-distributed magnetic field on all three axes, we need to address a key question: *how to place the coil for each sensor?*

For each magnetometer, the observed magnetic field is:

$$B^{coil} = \frac{\mu_0}{4\pi} \left(\frac{3(m \cdot r)r}{|r|^5} - \frac{m}{|r|^3} \right). \tag{12}$$

Since $|B^{coil}|$ changes linearly with $|r|$ and $|m|$, we use normalized r and m to simplify the problem:

(1) $r = [\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta]^T$
(2) $m = [\sin\beta\cos\gamma, \sin\beta\sin\gamma, \cos\beta]^T$,

where θ, ϕ are the position of the electromagnet in sphere coordinates, β, γ are the orientation of the electromagnet.

According to Eq. (12), the sensor will observe a magnet field parallel to the direction from the electromagnet to the sensor when the North pole of the electromagnet is pointing towards the sensor. Therefore, given $\theta = \beta, \phi = \gamma, B_x : B_y : B_z = \sin\theta\cos\phi : \sin\theta\sin\phi : \cos\theta$, we can have $B_x : B_y : B_z = 1 : 1 : 1$ by setting $\theta = 54.73°, \phi = 45° -$ the optimal alignment for our coils.

A compact design of the coil's form factor (e.g., diameter and length) is essential for MAGIC's usability. According to Biot-Savart law [19], the magnetic field strength is inversely proportional to the distance cubed. To enable a small form factor, we install a small coil for each magnetometer in the sensing array. According to the Ampere's law, this design also provides a sufficient magnetic field with minimal electric current. Our empirical study used two coil sizes (dimension, length) = (1.4 mm, 6.57 mm) and (0.7 mm, 4.45

mm). As shown in our experimental results in Sec. 7, a smaller coil size does not undermine the calibration performance. In MAGIC, we use the coil design as shown in Fig. 6 A.

We construct a 3D-printed stand to facilitate the form factor and stabilize the coil. Fig. 6A shows the coil stand. To stabilize the coil, we first insert it through the internal diagonal opening of the stand. Next, we glue the stand on the tracking platform. One can also stabilize the stand with strong glue (e.g., epoxy), achieving maximum stability of the coil. Our final assembly of the coil design is shown in Fig. 6B. Note that the assembly ensures the coil orientation, $\theta = 54.73°, \phi = 45°$. By miniaturizing the coil and stand, we can further shrink the size of the assembly for integration into compact magnetic sensing platforms.

6.2 Calibrating Signal and Driving Circuit

According to the soft-iron calibration scheme, we employ a rectangular signal for driving the proposed coils. Specifically, we control the current with a constant-current circuit design. The driving circuit receives the control signal from the micro-control unit (MCU) and generates a constant current for all coils. Fig. 7 shows its schematics. LT3055EDE [1] is a linear regulator with an internal protection circuitry. It can protect MCU from the reverse current in the coil circuit.

We examine the magnetic field patterns induced by different current settings. Specifically, we tested four current settings, i.e., 63 mA, 110 mA, 164 mA, and 293 mA. As shown in Fig. 8, although all current settings can effectively induce a noticeable magnetic field, a lower current may lead to fluctuations in the magnetic field. So, we choose 110 mA to generate the calibration signal.
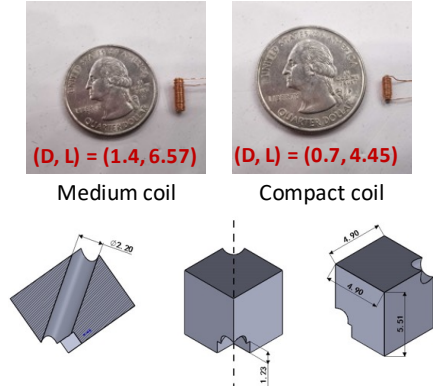
7 EVALUATION OF MAGIC

We chose MLX90393 chip [5] as it is a representative MEMS magnetometer for different magnetic sensing apps [10, 13]. We built a magnet tracking platform based on the open-source hardware of MagX [13]. The platform consists of 8 magnetometers on two parallel planes and a Bluetooth module for data communication. The sensing array collects uncalibrated raw data and sends the data via BLE to the computing unit, e.g., laptops or Raspberry Pi, for a subsequent analysis. We set the sample rate of the magnetic field sensor to 16 Hz. We use a Thinkpad X1 (with an Intel i5 chip) as the computing unit. We recruited 3 participants for our experimental evaluation.

We use the tracking accuracy as the metric. To collect the ground-truth data of the moving trajectory, we used the Leap motion-based platform, following the methods in MagX [13]. The tracking accuracy of Leap motion is within 2.5 mm [47].

7.1 Hard-iron Calibration Performance

7.1.1 Performance of hard-iron calibration. In this experiment, we first rotated the sensor array thoroughly (approximately for 30s) without any nearby magnet. The thus-collected data was used for the full-sphere calibration. Next, we extracted the first three seconds of data as inadequate calibration data. Then, we placed the platform at a stationary position for about one second and collected the data (i.e., 16 data points) for MAGIC's hard-iron calibration. Finally, we moved the magnet and collected magnetometer readings, while

A. Design of the coil and stand

(D, L) = (1.4, 6.57)   (D, L) = (0.7, 4.45)

Medium coil   Compact coil

B. MAGIC's assembly on MagX system

3D-printed coil stand
Solenoid coil
COTS magnetometer
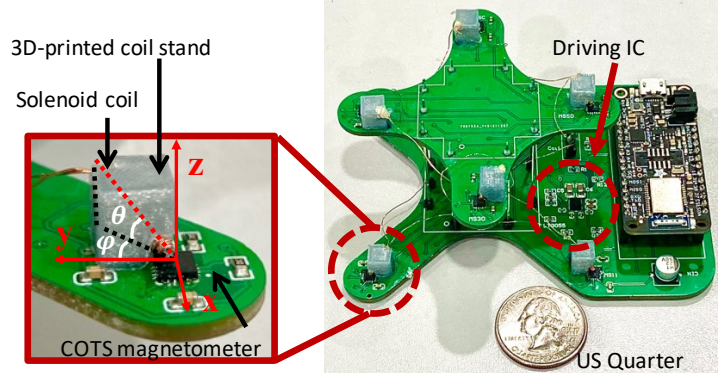Driving IC
US Quarter

Figure 6: Design and application of MAGIC's soft-iron calibration hardware on magnetic system. A. The configurations of coil and 3-D printed stand (cutaway and two axonometric views); B. The final assembly of MAGIC on MagX [13] system. Unit for the length is millimeter (mm), the assembly ensures $\theta = 54.73°$, $\phi = 45°$.
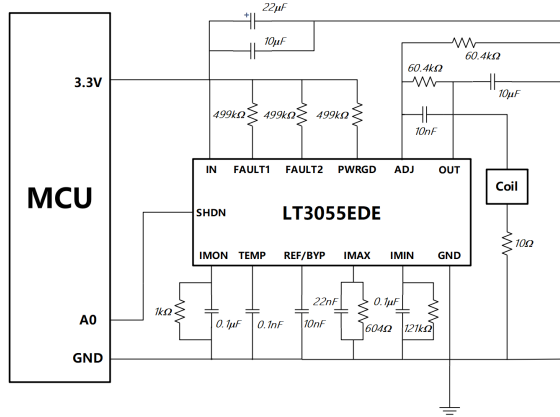

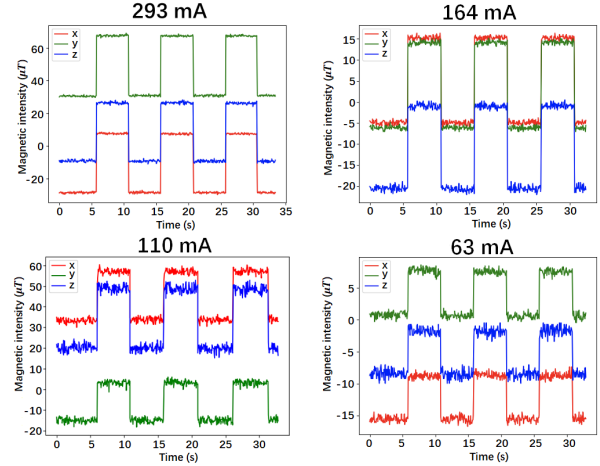**Figure 7: The schematics for coils' driving circuit.**

Figure 8: Rectangular magnetic field controlled by our coil circuit with varying current settings.

recording the magnet's actual trajectory using Leap Motion. This step was repeated three times.

To eliminate the soft-iron effect on tracking performance, we adopt the same soft-iron parameters, which are calculated with full-sphere calibration data by the sphere-fitting algorithm, for all the calibration methods. The hard-iron parameters of full-sphere calibration and inadequate calibration are also calculated by the sphere-fitting algorithm, with full-sphere calibration data and inadequate calibration data, respectively. The hard-iron parameters of MAGIC are the average of "one-second data".

Fig. 9 a) shows the tracking performance of 3 different calibration methods. The tracking error of inadequate calibration is 1.8*cm* greater than the full calibration on average. MAGIC can calibrate the hard-iron effect of magnetometers in 1 second. Note that MAGIC's hard-iron calibration module even outperforms the lengthy full-sphere movement. Specifically, the tracking accuracy is 9.16%, 13.19%, and 23.62% lower than the conventional method at 13 cm, 21 cm, and 27 cm distances, respectively.

7.1.2 Impact of # of points for hard-iron calibration. How many data points are sufficient for hard-iron calibration? To answer this question, we randomly sampled one data point from 16 points as MAGIC's hard-iron calibration input. We then changed the number of samples to examine the performance with different # of data points. We repeated this process seven times.

Fig. 10 shows the result with the distance of 20–22 cm between the magnet and the sensor array. The hard-iron calibration performance converges with *only* 2–4 data points. Note that with just one data point, MAGIC's hard-iron calibration can still achieve a sub-centimeter accuracy, indicating MAGIC's effective and rapid hard-iron calibration.

7.2 Soft-iron Calibration Performance.

7.2.1 Soft-iron settings. We installed our coil assembly on the MagX platform as described in Sec. 6. To change the soft-iron environment, we use soft-iron cylinders (i.e., ferrite rod [4, 18])
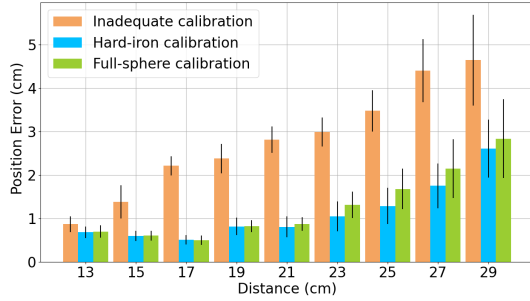
(a) Performance with hard-iron calibration.
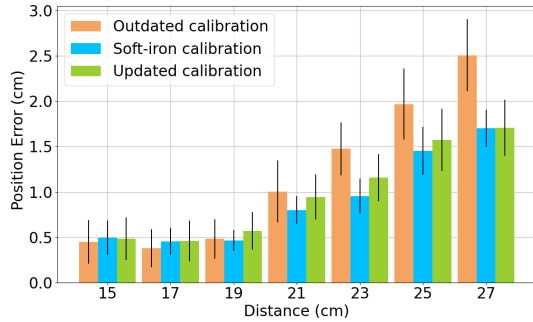
(b) Performance with soft-iron calibration.

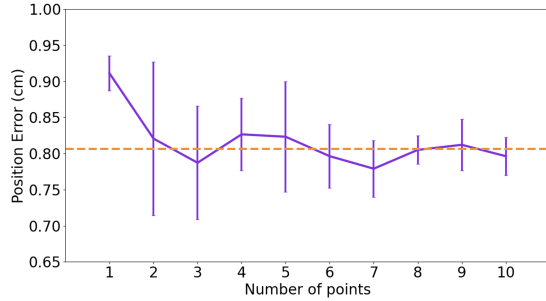**Figure 9: MAGIC's calibration performances.**

Figure 10: Hard-iron calibration with varying # of sample points. The dashed line shows the tracking result with all hard-iron calibration points.

which are made of magnetically permeable materials, widely used in electronic devices. This material generates an induced magnetic field proportional to the ambient field. This induced field will distort the nearby magnetic field, thus changing the soft-iron environment. The cylinders are placed randomly near our tracking platform to emulate the soft-iron disturbance.

7.2.2 The performance of soft-iron calibration. In this set of experiments, we compare the performance of three different calibration schemes in the presence of soft-iron disturbance. These calibration methods are: (1) "Outdated calibration": calibrate the sensor array without any soft-iron disturbance; (2) "Soft-iron calibration": calibrate the system with the MAGIC; and (3) "Updated calibration":

calibrate the sensor array with the full-sphere method. We repeated this process three times.

The tracking performance is shown in Fig. 9(b). We have two key findings: (1) the updated calibration method outperforms the outdated method; (2) despite the heavy soft-iron disturbance, MAGIC outperforms the updated approaches in most cases. The first finding indicates that in the presence of soft-iron disturbance, an uncalibrated sensor has the worst performance. The second finding suggests that MAGIC can achieve better performance than the full-sphere calibration. Compared to the full-sphere method, at 19cm, 21cm, and 23cm distances, MAGIC exhibits 19.0%, 15.5%, and 17.7% lower error rates, respectively. Note that MAGIC is worse than the outdated calibration in near field. This is due to the approximation error of the dipole model. The dipole model is applicable only when the distance between the magnet and the sensor is much greater than the radius of the sphere magnet, thus increasing the tracking error when the magnet is in the near field of the platform.

7.3 Performance of Automatic Triggering

We now introduce how to choose the hyperparameters of the diversity checker, and evaluate the accuracy of the triggering module. The hyperparameter setting varies with application scenarios and thus should be determined before using the auto-triggering scheme.

7.3.1 Tuning Hyperparameters. As mentioned in Sec. 5.1, we use a diversity checker to collect data for baseline estimation. This module has two hyperparameters: the threshold and the buffer size, which should be chosen according to the specific usage scenario. We will elaborate how to set the hyperparameters using the hand-tracking app as an example.

We first log all the possible platform movements during normal usage and employ a grid search to find the best combination of hyperparameters. During the data-collection process, the user wears the sensing platform on his/her wrist to execute the hand-tracking app. Note that wrist movements can be divided into elemental motions: horizontal, vertical, forward/backward, and rotation. The users are instructed to perform 5 movements individually: figure 8-shaped and four elemental motions.

We repeat each gesture five times. Then, we apply different combinations of the hyperparameters (i.e., distance threshold and buffer size) defined in Sec. 5.1. We recruited another participant for the experiment in a different lab setting. Our experimental results are summarized in Fig. 11.

Threshold setting. The result are shown in Fig. 11(a) with violin plot. We compare the average fitting error for all the gestures and buffer sizes. The error decreases as the threshold gets larger and converges after the threshold gets larger than $9\mu T$. The convergence hints "sufficient" data for the sphere-fitting algorithm.

Buffer size. The experimental results are shown in Fig. 11(b). Specifically, when the buffer is large, the error of all movements converges below the sensor noise floor, However, a large buffer may lead to a lengthy data collection phase. The fitting errors increase as the buffer size decreases, thus degrading the detection accuracy.

Given the above findings, we test the impact of different gestures by setting buff size and threshold to 300 and 9, respectively. By applying our setting, the mean fitting error decreases to about $2\mu T$ and the 75 percentile for each gesture are located below the sensor
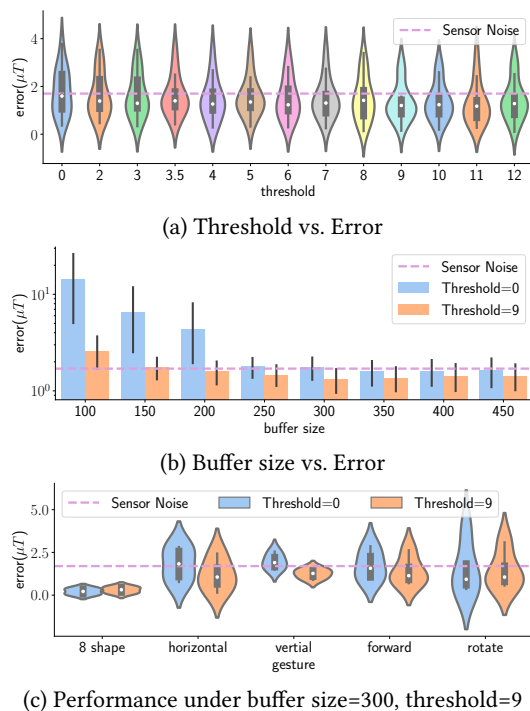
(a) Threshold vs. Error

(b) Buffer size vs. Error

(c) Performance under buffer size=300, threshold=9

Figure 11: Comparison of baseline setting accuracy.

	precision	recall	f1-score	support
normal	0.90	0.78	0.84	272
soft/hard-iron	0.80	0.91	0.85	253
accuracy			0.84	525
macro avg	0.85	0.85	0.84	525
weighted avg	0.85	0.84	0.84	525
(a) SVM				
	precision	recall	f1-score	support
normal	0.74	0.79	0.76	258
soft/hard-iron	0.76	0.71	0.74	252
accuracy			0.75	510
macro avg	0.75	0.75	0.75	510
weighted avg	0.75	0.75	0.75	510
(b) Proposed method				

Table 1: Classification result of detection stage.

Interval	auto-triggering	300 s	150 s	30 s
E(J)	270.82	270.82	275.24	310.59
Occ	3	3	6	30

Table 2: Overheads of periodic and auto triggering schemes. E and Occ denote the energy consumption and # occurrences of calibrations, respectively.

noise of MLX90393 magnetometer [5], indicating that t he error is "small" for the system.

7.3.2 Performance. We collect two types of data for evaluation of the detection algorithm. First, we collect the data without varying soft/hard-iron disturbance. The participant was instructed to use the sensing platform as usual after manually setting the baseline by using 8-shaped movements. The platform is re-calibrated every 30 seconds to ensure no soft/hard-iron change in the detection stage. Throughout this experiment, we collected 5 features as defined in Sec. 5.2. All collected data were labeled as "no need to re-calibrate". The second experiment followed the same protocol. The major difference is: we manually added the soft/hard-iron disturbance to the platform after finishing the baseline setting. The collected data in this experiment were labeled as "need re-calibration".

We conducted experiments with three different placements and randomly placed the small magnets within 20 cm of the platform. We use three types of magnets: one magnet, two stacked magnets, and three stacked magnets to emulate the magnetic disturbances in the real-world. The performance of our proposed algorithm is compared with the SVM algorithm [21]. We use grid search [42] to find the optimal parameter set of SVM. As shown in Table 1(a), the SVM can achieve 85% average F-1 score using radial basis function (RBF) kernel with $C = 1000, \gamma = 0.001$. Table 1(b) shows the performance of our proposed method. Specifically, our method achieves a 75% average F-1 score, 10% less than SVM. This method only incurs linear time complexity, i.e., $O(N)$. It is more efficient than SVM, which has a cubic time complexity, i.e., $O(N^3)$ [8].

7.3.3 Periodic vs. Auto-triggering. The environmental magnetic disturbances often incur arbitrary magnetic noises in tracking apps.

Thus, periodic calibration with a fixed time interval is unsuitable for real-world apps. Specifically, less frequent calibration would leave the error unattended while high-frequency calibration can incur unnecessary battery drain and computational overhead. We empirically compare auto-triggering with periodic calibration.

In our experiments, a magnet is moved repetitively along a pre-determined trajectory for 15 minutes. Specifically, the path is a semi-circular arc with a radius of 12cm. The sensing platform is placed at the center of the arc. After a 70s experiment, we place a small magnet next to the sensing platform. This step introduces a disturbance, a calibration is necessary for high-accuracy tracking. After 760s of the experiment, we move the small magnet away. This step removes the disturbance, thus needing another calibration to adapt to the new environment. Without calibration, this varying magnetic environment would severely undermine the tracking performance.

Our auto-triggering detects the disturbance and re-calibrates the platform in a timely manner. The periodic method, on the other hand, is to calibrate the system at a fixed interval. In our experiments, three different periods — 30, 150, and 300 seconds — are selected for emulating short, medium, and long intervals, respectively. Fig. 13 shows our tracking results. Compared to the periodic method, the auto-triggering method can help achieve accurate and robust tracking performance. For the periodic method, a long time interval would severely undermine the tracking accuracy. Specifically, at a 300s interval, more than 40% of results show greater than 3cm tracking error. On the other hand, frequent periodic calibrations induces unnecessary energy consumption. As shown in Table 2, the periodic method with an interval of 30s consumes 15% more energy than the auto-triggering.

Electronic devices tested in experiments	No disturbance	Smart wristband	Charging case of wireless earbuds	Wireless earbud	Wireless keyboard	Wireless mouse	Bone conduction earphone	Robot arm
Need calib.?	NO	YES	YES	NO	NO	NO	YES	NO
Err. (mean, std)	(0.98, 0.04)	(6.52, 0.78)	(6.76, 0.46)	(2.73, 0.11)	(0.98, 0.03)	(1.01, 0.04)	(9.11, 6.88)	(1.11, 0.13)

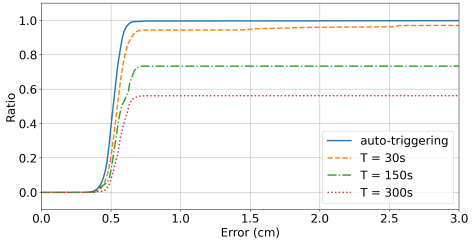**Figure 12: The detection result of the auto-triggering module. The unit for position errors is cm.**

Figure 13: The CDF of periodic and auto triggering. T denotes different intervals of periodic tracking.

7.4 Applicability of MAGIC

So far, we have evaluated the tracking performance by using the analytical method, i.e., dipole model-based. MAGIC can improve magnetic tracking performance regardless of the choice of the tracking algorithm. To corroborate this, we examine MAGIC's performance with other magnetic tracking algorithms, i.e., machine learning (ML) and the Kalman filter (KF).

7.4.1 ML Method. First, we pre-process the data by demeaning (i.e., subtracting the mean value) sensor readings. Then, we trained an extra tree regressor model [22] with 200 estimators. We divide data into 70% training and 30% test sets. In the training stage, we adopt five-fold cross-validation [50]. The data used in the training stage is calibrated using the full-sphere calibration method. In the testing stage, we process the data with the full-sphere method and MAGIC individually. As shown in Fig. 14, compared to the full-sphere method, MAGIC achieves similar mean tracking error and smaller variance. This result indicates MAGIC can improve the ML-based method, thus extending its application scope.

7.4.2 Unscented Kalman Filter (UKF) method [11]. We have implemented the magnetic tracking method based on UKF [45] which models the tracking process as a KF process. The state (position, orientation) of the magnet is used as the filter's state, and the sensor reading on each magnetometer is used as the filter's observation. The UKF tracking consists of two stages: prediction and update. In the prediction stage, the filter predicts the state for the next timestamp by using the state-transition function (usually the equations of motion). In the update stage, the filter adjusts its state prediction according to the observation function, i.e., the sensor reading equation (Eq. (1)). In our experiment, we use the first-order motion equation as the state-transition function. We apply the initial state of the filter as the ground truth and use the sensor noise as the observation noise. We compare the tracking performance using
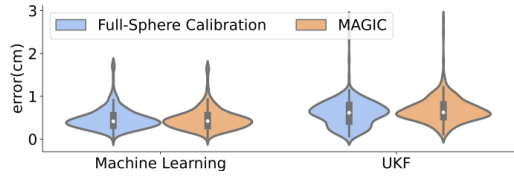

Figure 14: Performance of different tracking methods.

both full-sphere calibration method and MAGIC. As shown in Fig. 14, MAGIC achieves a lower tracking error than the legacy method.

7.5 Energy Consumption

We now analyze MAGIC's energy consumption. Our hard-iron calibration (Sec. 4.2) incurs minimal energy overhead thanks to its linear time complexity design. In our soft-iron experiment, the current, voltage, and power for driving our calibration circuit are 110 mA, 0.1 V, and 11 mW, respectively. Since the soft-iron calibration only needs 1s, the energy consumption of each calibration is 0.011J. We use a Raspberry Pi 4B as the computing platform and utilize a power meter to measure energy consumption. Compared to the tracking task, the extra energy consumption of the computing platform of MAGIC is 0.097J. Note that the energy consumption of the stand-alone tracking task is 2.53J in 1s. Hence, MAGIC's calibration incurs only 4.27% extra power overhead.

8 MAGIC IN ACTION

In what follows, we will first study how real-world disturbances would impact MAGIC. Then, we use the free-form writing as a use-case to demonstrate high-precision tracking even in the presence of environmental noises.

8.1 Disturbance of COTS Devices

We investigate the impact of some common electronic devices — such as the smart wristband, earbuds, wireless keyboard, wireless mouse, etc. — on the sensing platform. We measure the impact of these devices on tracking accuracy. First, we fixed the magnet at the same position (i.e., $x = 12cm, y = 13cm$, and $z = -0.1cm$). Then, we attached electronic devices near the platform to emulate real-world usages. For each device, we recorded the error rate for positioning the magnet. We also recorded the result (i.e., "Need Calibration" or "No Calibration is Needed") of the detection module. We conducted an experiment with the parameter $k_3 = 3.5$ and summarized the results in Fig. 12. The auto-triggering scheme can automatically detect if calibration is need in the presence of real-world disturbances. The actual tracking performance reflected the detection
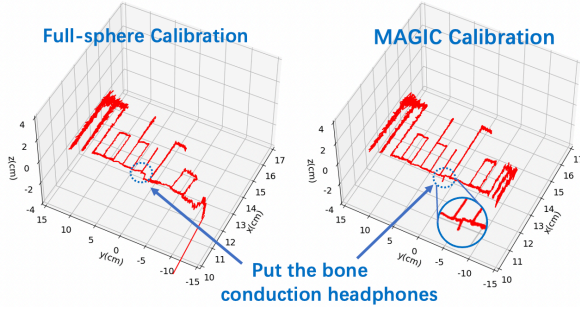
Figure 15: The tracking trajectory calibrated with the full-sphere method and MAGIC.

Figure 16: Error in moving trajectory between MAGIC and full-sphere calibration.

result. Specifically, the averaged position errors of "No Calibration is Needed" and "Need Calibration" conditions are 1.36 cm and 7.46 cm, respectively. The empirical results indicate that MAGIC can automatically control the error range of magnetic sensing app, thus enhancing the app's usability. Note that system designers can also tune the hyperparameters to adopt their system for different apps.

8.2 Free-form Writing Experiment

We have built a free-form writing app to demonstrate how MAGIC can help maintain the tracking accuracy in the presence of environmental changes. In this experiment, we draw a predetermined movement trajectory for the word "MobiCom" on a 2D surface. The trajectory is located on the plane of $z = -0.1$ cm from the platform coordinate system. Before starting to write, we performed the full-sphere calibration for 30s. Then, we put the platform on the table and run MAGIC. After keeping the platform stationary for 5s to obtain the calibration data, we placed a magnet on the table.

The platform was turned on at the same time to set the trigger parameters. Then, we moved the magnet slowly along the "MobiCom" trajectory. For now, we show the tracking performance in real time at 16Hz. The triggering results, i.e., "not need" or "NEED CALIBRATION", were shown at 2s intervals. If there is a "NEED CALIBRATION" message, we move the magnet away from the platform and then re-calibrate the system.

We emulate real-world disturbances in the app. Specifically, we put the bone conduction headphones (Shokz OpenMove AS660) near the platform, when the magnet moved between the letter "i" and "C". The auto-triggering module promptly showed the "NEED CALIBRATION" message. Next, we moved the magnet away to re-calibrate sensor readings. Finally, we resume the movement along the predetermined path.

As shown in Fig. 16, the tracking trajectory was severely distorted with the nearby headphones. For example, when the magnet moves to the last letter "m", the un-calibrated tracking error is more than 6 cm. MAGIC can correct the error, thus fixing the deformed trajectory. The quantitative results of magnet movement trajectory are plotted in Fig. 15. Specifically, after placing the headphones near the platform, the magnet trajectory starts to deviate. Without MAGIC, the tracking platform will not be able to notice the hard/soft-iron disturbances. As a result, the tracking error grew rapidly. In contrast, by detecting the disturbance and calibrating the platform, MAGIC can reduce the tracking error by automatically triggering
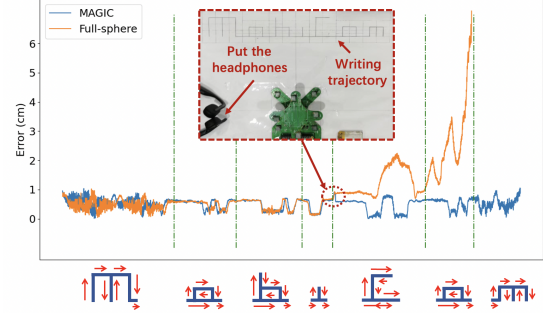
the calibration. This way, MAGIC can keep the error lower than 1 cm throughout the experiment.

8.3 Hand-writing Recognition

This experiment studies how MAGIC can improve a prevalent handwriting recognition task. For the letters, we used the single-stroke alphabet, i.e., 20 lower case English letters as shown in Fig. 17. The average size of each letter is about 8cm $\times$ 8cm. In the experiment, we asked 3 volunteers to write all letters freely next to the tracking platform. During the writing process of each letter, we used a permanent magnet to introduce disturbances. We compare the performance of the sphere-fitting and MAGIC calibration methods, respectively. For the sphere-fitting method, each participant performs a one-time calibration procedure before starting the writing session. In contrast, MAGIC calibration leverages the auto-triggering scheme. Participants were instructed to write each letter six times for each calibration method. We projected the 3D trajectory to the X-Y plane for the handwriting recognition task. We trained the ML classifier with the MNIST [16, 26] Letters dataset and ResNet [20].
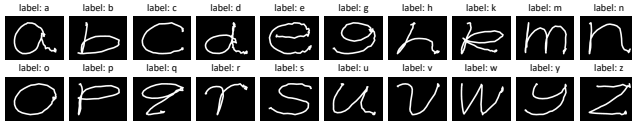
Fig. 17 compares the magnetic handwriting calibrated by MAGIC and sphere-fitting methods, respectively. As shown in Figs. 17 (a) and (b), MAGIC helps calibrate the writing trajectory. For the sphere-fitting approach, we observe a similar drifting issue discussed in Sec. 8.2. Based on the classification results, i.e., Fig. 17 (c) and (d), MAGIC can effectively reduce the effect of magnetic disturbance. Overall, MAGIC and sphere-fitting methods achieve 90.3% and 12.8% recognition accuracy, respectively. This empirical study demonstrates MAGIC's potential for improving essential applications.
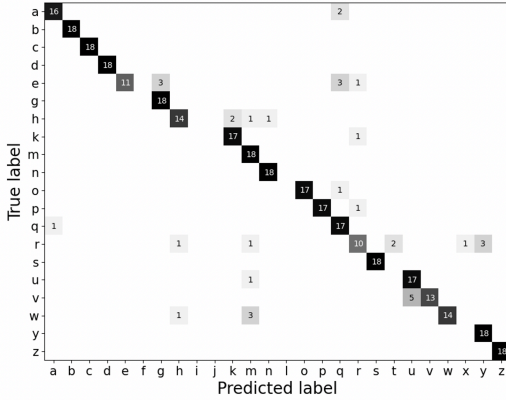
9 RELATED WORK

There are two major types of calibration: magnetometer-only and IMU-aided calibration.

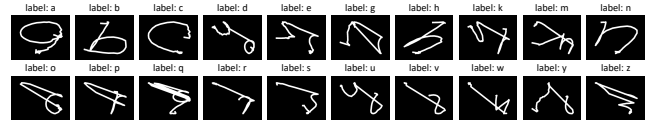## 9.1 Magnetometer-only Calibration

The latest Android OS [6] employs the Kasa fitting algorithm [23] for calibration. This is an efficient sphere-fitting algorithm that calculates the center and the radius of a sphere by solving a system of linear equations. It has a linear time complexity and is thus power-efficient. However, it does not address the soft-iron effect. As shown in Sec. 7.2, ignoring soft-iron disturbances can significantly degrade the performance of high-accuracy magnetic sensing.
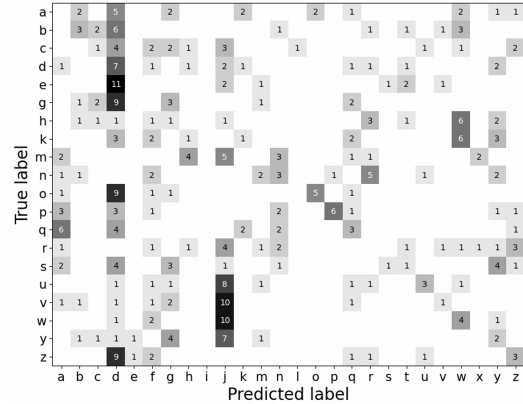
label: a label: b label: c label: d label: e label: g label: h label: k label: m label: n

label: o label: p label: q label: r label: s label: u label: v label: w label: y label: z

(a) With MAGIC calibration

label: a label: b label: c label: d label: e label: g label: h label: k label: m label: n

label: o label: p label: q label: r label: s label: u label: v label: w label: y label: z

(b) With sphere-fitting calibration

(c) With MAGIC calibration

(d) With sphere-fitting calibration

Figure 17: (a) and (b) show the single-stroke alphabet magnetic writing, calibrated w/ MAGIC and sphere-fitting methods, respectively. Confusion matrices in (c) and (d) present the performance of handwriting classification tasks.

Zhang *et al.* [49] proposed a calibration method using ellipsoid fitting. This algorithm can derive both soft-iron effect on each axis (3-DoF) and the hard-iron parameter. However, this approach only considered the soft-iron on each axis, but failed to address the soft-iron anomaly problem.

9.2 IMU-aided calibration

Papafotis *et al.* [33] presented a unified calibration methodology for both magnetometer and motion sensors (e.g., gyroscope and accelerometer). The soft/hard-iron parameters are calculated iteratively using the gradient descent method. It can achieve 6-DoF soft-iron and hard-iron calibration.

Wu *et al.* [48] proposed a calibration algorithm based on IMU and magnetometer. Instead of solving for the calibration parameter with the collected data, they used an extended Kalman filter (EKF) to update the estimation of the soft/hard-iron parameter dynamically. They built a dynamic transition function between two consecutive data according to rotation and acceleration. This method can calculate 6-DoF soft-iron and hard-iron parameters.

There are three key findings. First, using IMU data can help reduce the calibration effort, but incurs more energy cost. Second, the user has to rotate the platform extensively in order to calibrate the sensor. Third, all of the above algorithms are based on the presumption that the sensor reading lies on a unit sphere, making it susceptible to the soft-iron anomaly. Compared to the existing methods, MAGIC is power-efficient, user-friendly (no need to rotate the platform), and is resilient to the soft-iron anomaly.

10 DISCUSSIONS

MAGIC can be integrated with different form factors of magnetic tracking platforms. For example, in Figs. 6 and 12, the sensing platform was adopted from MagX [13], an open-sourced magnetic tracking system. Smaller tracking arrays can also use MAGIC. Specifically,

MAGIC can calibrate the soft-iron effect for the single-magnetometer setting. For correcting the hard-iron effect, MAGIC needs the configuration of the environmental magnetic field. To solve Eq. (9) with LM algorithm, MAGIC needs at least three magnetometers to fix the hard-iron effect. Meeting this requirement can also enable the detection phase of MAGIC's auto-triggering.

The design of MAGIC can facilitate a smaller coil size. As shown in Sec. 6, the soft-iron calibration requires a noticeable controlled magnetic field. That is, one can further reduce the size as the coil can provide a sufficient magnetic field, e.g., about 30 μT as shown in Fig. 8. For example, by enabling a built-in MEMS magnetic inductor [25], one can construct an even smaller form factor.

11 CONCLUSION

We present MAGIC, a practical and easy-to-use calibration system for magnetic sensing apps. MAGIC handles notorious soft- and hard-iron disturbances with a novel approach. By using a lightweight auto-triggering scheme, MAGIC can also enable end-to-end automatic calibration. We have corroborated MAGIC's performance in real-world apps with a series of empirical studies. Specifically, MAGIC's hard- and soft-iron calibration is shown to outperform state-of-the-arts with much less user involvement. MAGIC is expected to pave the way of real-world apps of magnetic sensing technology.

ACKNOWLEDGEMENTS

REFERENCES

[1] 2013-2018. 500mA, Linear Regulator with Precision Current Limit and Diagnostics. https://www.mouser.com/datasheet/2/609/LT3055-1504143.pdf. (2013-2018).

[2] 2015. LSM9DS1 Datasheet, ST Microelectronics. https://www.st.com/resource/en/datasheet/lsm9ds1.pdf. (2015).

[3] 2017. Android Tutorial on Calibrating Phone or Tablet. https://support.google.com/maps/answer/2839911?hl=en&visit_id=637750697657655679-1164089685&rd=1&co=GENIE.Platform%3DAndroid&oco=1#zippy=%2Cimprove-accuracy-of-your-location-with-live-view%2Ccalibrate-your-phone-or-tablet. (2017).

[4] 2017. Magnetic Core. https://en.wikipedia.org/wiki/Magnetic_core. (2017).

[5] 2017. Melexis: MLX90393 Triaxis® Magnetic Node. https://www.melexis.com/-/media/files/documents/datasheets/mlx90393-datasheet-melexis.pdf. (2017).

[6] 2021a. The Android Codebase for Calibration. https://cs.android.com/android/platform/superproject/+/master:device/google/contexthub/firmware/os/algos/calibration/. (2021).

[7] 2021b. The Android Codebase for diversity checker. https://cs.android.com/android/platform/superproject/+/master:device/google/contexthub/firmware/os/algos/calibration/diversity_checker/. (2021).

[8] Abdiansah Abdiansah and Retantyo Wardoyo. 2015. Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM. *International Journal of Computer Applications* 128, 3 (2015), 28–34. DOI:http://dx.doi.org/10.5120/ijca2015906480

[9] Sudhakar A.N., Markandeya R., Srinivasa Rao B., Kumar Pandey Ajoy, and Kaushik D. 2021. Effect of alloying elements on the microstructure and mechanical properties of high chromium white cast iron and Ni-Hard iron. *Materials Today: Proceedings* (2021). DOI:http://dx.doi.org/https://doi.org/10.1016/j.matpr.2021.10.284

[10] Raunaq Bhirangi, Tess Hellebrekers, Carmel Majidi, and Abhinav Gupta. 2021. ReSkin: Versatile, Replaceable, Lasting Tactile Skins. In *5th Annual Conference on Robot Learning (CoRL)*.

[11] Marius Birsan. 2011. Recursive Bayesian method for magnetic dipole tracking with a tensor gradiometer. *IEEE Transactions on Magnetics* 47, 2 PART 2 (2011), 409–415. DOI:http://dx.doi.org/10.1109/TMAG.2010.2091964

[12] Jean G. Van Bladel. 2007. *Electromagnetic Fields* (2 ed.). Wiley-IEEE Press.

[13] Dongyao Chen, Mingke Wang, Chenxi He, Qing Luo, Yasha Iravantchi, Alanson Sample, Kang G. Shin, and Xinbing Wang. 2021. MagX: Wearable, Untethered Hands Tracking with Passive Magnets. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking (MobiCom '21)*. Association for Computing Machinery, New York, NY, USA, 269–282. DOI:http://dx.doi.org/10.1145/3447993.3483260

[14] Ke-Yu Chen, Kent Lyons, Sean White, and Shwetak Patel. 2013. UTrack: 3D Input Using Two Magnetic Sensors. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology (UIST '13)*. Association for Computing Machinery, New York, NY, USA, 237–244. DOI:http://dx.doi.org/10.1145/2501988.2502035

[15] Ke-Yu Chen, Shwetak N. Patel, and Sean Keller. 2016. Finexus: Tracking Precise Motions of Multiple Fingertips Using Magnetic Sensing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, New York, NY, USA, 1504–1514. https://doi.org/10.1145/2858036.2858125

[16] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. 2017. EMNIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*. 2921–2926. DOI:http://dx.doi.org/10.1109/IJCNN.2017.7966217

[17] R.P. Feynman, R.B. Leighton, M. Sands, and EM Hafner. 1965. *The Feynman Lectures on Physics*. Vol. 33. AAPT. 750 pages.

[18] Rudolf F. Graf. 1999. S. In *Modern Dictionary of Electronics (Seventh Edition)* (seventh edition ed.), Rudolf F. Graf (Ed.). Newnes, Boston, 662–758. DOI:http://dx.doi.org/https://doi.org/10.1016/B978-0-08-051198-6.50024-2

[19] Ian S Grant and William Robert Phillips. 2013. *Electromagnetism*. John Wiley & Sons.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. *CoRR* abs/1512.03385 (2015). http://arxiv.org/abs/1512.03385

[21] Thorsten Joachims. 1998. *Making large-scale SVM learning practical*. Technical Report 1998,28. Dortmund. http://hdl.handle.net/10419/77178

[22] Vijay John, Nithilan Meenakshi Karunakaran, Chunzhao Guo, Kiyosumi Kidono, and Seiichi Mita. 2018. Free Space, Visible and Missing Lane Marker Estimation using the PsiNet and Extra Trees Regression. In *2018 24th International Conference on Pattern Recognition (ICPR)*. 189–194. DOI:http://dx.doi.org/10.1109/ICPR.2018.8546108

[23] I. Kasa. 1976. A Circle Fitting Procedure and its Error Analysis. *IEEE Transactions on Instrumentation and Measurement* IM-25, 1 (1976), 8–14. DOI:http://dx.doi.org/10.1109/TIM.1976.6312298

[24] David Kim, Otmar Hilliges, Shahram Izadi, Alex D. Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: Freehand 3D Interactions Anywhere Using a Wrist-Worn Gloveless Sensor. In *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology (UIST '12)*.

Association for Computing Machinery, New York, NY, USA, 167–176. DOI:http://dx.doi.org/10.1145/2380116.2380139

[25] Hoa Thanh Le, Rubaiyet I. Haque, Ziwei Ouyang, Seung Woo Lee, Shelley I. Fried, Ding Zhao, Min Qiu, and Anpan Han. 2021. MEMS inductor fabrication and emerging applications in power electronics and neurotechnologies. *Microsystems & Nanoengineering* 7, 1 (2021), 59. DOI:http://dx.doi.org/10.1038/s41378-021-00275-w

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324. DOI:http://dx.doi.org/10.1109/5.726791

[27] Yongqing Li, Peng Xiong, Stephan Von Molnár, Steffen Wirth, Yuzo Ohno, and Hideo Ohno. 2002. Hall magnetometry on a single iron nanoparticle. *Applied Physics Letters* 80, 24 (2002), 4644–4646.

[28] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar. *ACM Trans. Graph.* 35, 4, Article 142 (July 2016), 19 pages. DOI:http://dx.doi.org/10.1145/2897824.2925953

[29] Kent Lyons. 2020. Wearable Magnetic Field Sensing for Finger Tracking. In *Proceedings of the 2020 International Symposium on Wearable Computers (ISWC '20)*. Association for Computing Machinery, New York, NY, USA, 63–67. DOI:http://dx.doi.org/10.1145/3410531.3414304

[30] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. 2014. Gyrophone: Recognizing speech from gyroscope signals. *Proceedings of the 23rd USENIX Security Symposium* (2014), 1053–1067.

[31] André Michaud. 2013. On the magnetostatic Inverse cube law and magnetic monopoles. *International Journal of Engineering Research and Development e-ISSN* (2013), 50–66.

[32] J J More. 1977. Levenberg–Marquardt algorithm: implementation and theory. (1 1977).

[33] Konstantinos Papafotis and Paul P. Sotiriadis. 2019. MAG.I.C.AL.-A Unified Methodology for Magnetic and Inertial Sensors Calibration and Alignment. *IEEE Sensors Journal* 19, 18 (2019), 8241–8251. DOI:http://dx.doi.org/10.1109/JSEN.2019.2919179

[34] Feini Qu, Brendan D. Stoeckl, Peter Gebhard, Todd J. Hullfish, Josh R. Baxter, and Robert Leon Mauck. 2018. A Wearable Magnet-Based System to Assess Activity and Joint Flexion in Humans and Large Animals. *Annals of Biomedical Engineering* 46 (2018), 2069–2078.

[35] R. Racz, C. Schott, and S. Huber. 2004. Electronic compass sensor. In *SENSORS, 2004 IEEE*. 1446–1449 vol.3. DOI:http://dx.doi.org/10.1109/ICSENS.2004.1426458

[36] Pavel Ripka. 1992. Review of fluxgate sensors. *Sensors and Actuators A: Physical* 33, 3 (1992), 129–141. DOI:http://dx.doi.org/https://doi.org/10.1016/0924-4247(92)80159-Z

[37] W. M. Saslow. 1991. How a superconductor supports a magnet, how magnetically "soft" iron attracts a magnet, and eddy currents for the uninitiated. *American Journal of Physics* 59, 1 (1991), 16–25. DOI:http://dx.doi.org/10.1119/1.16700

[38] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. 2015. Accurate, Robust, and Flexible Real-Time Hand Tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3633–3642. DOI:http://dx.doi.org/10.1145/2702123.2702179

[39] Sheng Shen, Mahanth Gowda, and Romit Roy Choudhury. 2018. Closing the Gaps in Inertial Motion Tracking. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking (MobiCom '18)*. Association for Computing Machinery, New York, NY, USA, 429–444. DOI:http://dx.doi.org/10.1145/3241539.3241582

[40] Sheng Shen, He Wang, and Romit Roy Choudhury. 2016. I Am a Smartwatch and I Can Track My User's Arm. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '16)*. Association for Computing Machinery, New York, NY, USA, 85–96. DOI:http://dx.doi.org/10.1145/2906388.2906407

[41] Janez Stepišnik. 2006. NMR down to Earth. *Nature* 439, 7078 (2006), 799–800.

[42] Iwan Syarif, A. Prugel-Bennett, and Gary Wills. 2016. SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. *TELKOMNIKA (Telecommunication Computing Electronics and Control)* 14 (12 2016), 1502. DOI:http://dx.doi.org/10.12928/telkomnika.v14i4.3956

[43] C. R. Taylor, H. G. Abramson, and H. M. Herr. 2019. Low-Latency Tracking of Multiple Permanent Magnets. *IEEE Sensors Journal* 19, 23 (2019), 11458–11468. DOI:http://dx.doi.org/10.1109/JSEN.2019.2936766

[44] C. R. Taylor, S. S. Srinivasan, S. H. Yeon, M. K. O'Donnell, T. J. Roberts, and H. M. Herr. 2021. Magnetomicrometry. *Science Robotics* 6, 57 (2021), eabg0656. DOI:http://dx.doi.org/10.1126/scirobotics.abg0656

[45] Eric A Wan and Rudolph Van Der Merwe. 2000. The unscented Kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. Ieee, 153–158.

[46] Jue Wang, Deepak Vasisht, and Dina Katabi. 2014. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals. *SIGCOMM Comput. Commun. Rev.* 44, 4 (Aug. 2014), 235–246. DOI:http://dx.doi.org/10.1145/2740070.2626330

[47] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. 2013. Analysis of the accuracy and robustness of the Leap Motion Controller. *Sensors (Switzerland)* 13, 5 (2013), 6380–6393. DOI:http://dx.doi.org/10.3390/s130506380

[48] Yuanxin Wu, Danping Zou, Peilin Liu, and Wenxian Yu. 2018. Dynamic Magnetometer Calibration and Alignment to Inertial Sensors by Kalman Filtering. *IEEE Transactions on Control Systems Technology* 26, 2 (2018), 716–723. DOI:

http://dx.doi.org/10.1109/TCST.2017.2670527

[49] Zhang Xiaoming and Gao Lizhen. 2009. A novel auto-calibration method of the vector magnetometer. *ICEMI 2009 - Proceedings of 9th International Conference on Electronic Measurement and Instruments* (2009), 1145–1150. DOI:http://dx.doi.org/10.1109/ICEMI.2009.5274904

[50] Sanjay Yadav and Sanyam Shukla. 2016. Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification. In *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. 78–83. DOI:http://dx.doi.org/10.1109/IACC.2016.25