MechSense: A Design and Fabrication Pipeline for Integrating Rotary Encoders into 3D Printed Mechanisms

Marwa Alalawi
malalawi@mit.edu
MIT, CSAIL
USA

Noah Pacik-Nelson
Accenture Labs
USA

Junyi Zhu
MIT, CSAIL
USA

Ben Greenspan
Accenture Labs
USA

Andrew Doan
MIT, CSAIL
USA

Brandon Wong
MIT, CSAIL
USA

Benjamin Owen-Block
MIT, CSAIL
USA

Shanti Mickens
MIT, CSAIL
USA

Wilhelm Schoeman
MIT, CSAIL
USA

Michael Wessely
wessely@mit.edu
MIT, CSAIL
USA

Andreea Danielescu
andreea.danielescu@accenture.com
Accenture Labs
USA

Stefanie Mueller
stefanie.mueller@mit.edu
MIT, CSAIL
USA

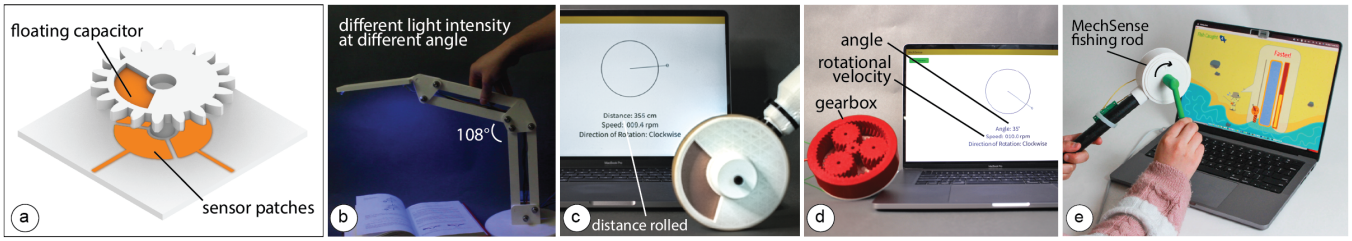

Figure 1: (a) MechSense allows 3D printed mechanisms to sense their direction of rotation, speed, and angular position using a floating capacitor on the moving part and sensor patches on the static part of the mechanism. (b) A smart linkage-based lamp that changes the intensity of the light when the top linkage bars is moved, (c) a distance measuring wheel that can measure the dimension of surfaces, (d) a planetary gear box that can sense its own state, and (e) a fishing rod shaped controller that interfaces with a 2D game.

ABSTRACT

We introduce *MechSense*, 3D-printed rotary encoders that can be fabricated in one pass alongside rotational mechanisms, and report on their angular position, direction of rotation, and speed. *MechSense* encoders utilize capacitive sensing by integrating a floating capacitor into the rotating element and three capacitive sensor patches in the stationary part of the mechanism. Unlike existing rotary encoders, *MechSense* does not require manual assembly but can be seamlessly integrated during design and fabrication. Our *MechSense* editor allows users to integrate the encoder with a rotating mechanism and exports files for 3D-printing. We contribute a sensor topology and a computational model that can compensate for print deviations. Our technical evaluation shows that *MechSense* can detect the angular position (mean error: 1.4°) across multiple prints and rotations, different spacing between sensor patches, and different sizes of sensors. We demonstrate *MechSense* through three application examples on 3D-printed tools, tangible UIs, and gearboxes.

CHI '23, April 23–28, 2023, Hamburg, Germany

ACM ISBN 978-1-4503-9421-5/23/04.
https://doi.org/10.1145/3544548.3581361

CCS CONCEPTS

• **Human-centered computing → Human computer interaction (HCI)**.

KEYWORDS

3D printed mechanisms, printed electronics, capacitive sensing.

ACM Reference Format:
Marwa Alalawi, Noah Pacik-Nelson, Junyi Zhu, Ben Greenspan, Andrew Doan, Brandon Wong, Benjamin Owen-Block, Shanti Mickens, Wilhelm Schoeman, Michael Wessely, Andreea Danielescu, and Stefanie Mueller. 2023. MechSense: A Design and Fabrication Pipeline for Integrating Rotary Encoders into 3D Printed Mechanisms. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems (CHI '23), April 23–28, 2023, Hamburg, Germany.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3544548.3581361

1 INTRODUCTION

Advances in 3D printing over the last decades have enabled increasingly complex 3D printed objects, including objects with moving parts that contain mechanisms, such as gears, linkages, and wheels (Grafter [15]). While early 3D printed mechanisms were fully passive, researchers have started exploring how to augment 3D printed mechanisms with sensors to enable interactive applications.

To sense the motion of these 3D printed mechanisms, researchers traditionally used external sensors. For instance, researchers have used acoustic sensing via external microphones (*Lamello* [16]) or used hall effect sensors and magnets (*MechaMagnets* [24]) to determine the interaction with 3D printed objects. Using external sensors, however, requires additional assembly, it can be difficult to augment mechanisms in the interior of a design, and many off-the-shelf sensors require to distribute wires throughout the 3D print as conductive filament often has a too high resistance, especially for larger prints.

Recently, the advent of multi-material 3D printing with conductive filament has enabled the integration of sensors with 3D printed geometries. For instance, researchers have shown how to print capacitive sensors from conductive filament to integrate touch sensors (*Capricate* [18]) and deformation sensors (MetaSense [5]) with 3D printed object geometries. Thus, a variety of different sensors can now be 3D printed together with the object geometry in one pass, facilitating the creation of interactive objects.

For rotational mechanisms, such as gears, linkages, and wheels, however, no sensor design has been proposed yet that can be 3D printed in one pass. To sense direction of rotation, speed, and angular position of rotational mechanisms, existing encoder designs require additional manual assembly. For instance, *SteelSense* [22] enables high-resolution sensing of rotational elements , but requires metal casting of the sensing elements that afterwards have to be manually integrated into a 3D printed casing. Similarly, Karali et al. [10] demonstrate a capacitive rotation sensor that utilizes two patterned copper plates that change their capacitance depending on the relative angle to each other. However, the copper plates cannot be 3D printed together with the object geometry, and thus, require manual assembly.

In this paper, we introduce *MechSense*, a fabrication pipeline based on conductive multi-material 3D printing that can print the geometry of the mechanisms and sensors together in one pass without the need to assemble the conductive and non-conductive parts. We focus on rotating mechanisms and develop a sensor layout that utilizes tracking of a floating capacitor which can augment various rotational mechanisms with the ability to sense their direction of rotation, speed, and angular position (Figure 1). To better understand user's familiarity with multi-material 3D-printing and to identify the design software they most commonly use, we conducted a survey of 20 hardware design professionals. Based on the survey results and the overall popularity of SolidWorks in industry, we built a 3D editor add-on for SolidWorks that automatically integrates the sensor layout into the mechanism's geometry and exports files for 3D printing. In our technical evaluation, we determine the angular position estimation error for sensors with different spacing between them, the effect of the sensor patch size, and the impact of the proximity of the user's hand near the *MechSense* encoder. We demonstrate its usefulness with three application examples for tangible user interfaces, construction tools, and gearboxes.

In summary, we contribute:

- a *sensor layout* based on a floating capacitor that can be integrated with 3D printable rotational mechanisms to determine their direction of rotation, speed, and angular position (mean error: 1.4°);
- a *3D editor extension* that automatically integrates sensors into mechanical components and generates the files for 3D printing;
- a *Java/Processing* tool that converts the raw sensor data into angular position, direction of rotation, and speed [1].
- a *technical evaluation* of the angular position accuracy for different spacing between sensors, the effect of the sensor patch size, and the influence of a user's hand in proximity to the capacitive sensor;
- four *applications* that demonstrate sensing integrated with various rotational mechanisms for tangible user interfaces, construction tools, and gearboxes.

2 RELATED WORK

Our work is related to research that investigates how to fabricate rotary encoders, 3D print sensors, and integrate sensors into mechanical elements.

2.1 Rotary Encoders Using Capacitive Sensing

Many rotational mechanisms use encoders to retrieve data on the position and the speed of the rotating element. Most commercial encoders utilize optical or inductive sensing approaches, but researchers have also investigated ways to fabricate capacitive encoders since the sensor elements do not require contact between the stationary sensor and the rotating element. For example, Cermak et al. [3], Gasulla et al. [4], Ferrari et al. [2], and Karali et al. [10] developed rotational encoders that consist of two stationary circular conductive plates opposite of one another, where one of the plates is segmented into electrodes that acts as a capacitive pair with the other stationary plate. The two stationary plates are separated by a rotating insulator or a conductive plate that triggers changes in the capacitance for each electrode. In all these approaches, the suggested encoder geometry consists of at least three plates that have to be individually fabricated, for example, by copper etching, and have to be manually mounted on a motor shaft. Additionally, Zheng et al. [25], Hou et al. [8] and Wang et al. [23] demonstrated approaches that utilize a pair of conductive circular plates, with one being a stator and one being a floating conductive rotor that is

[1] https://github.com/HCIELab/MechSense

segmented into two parts via a complex pattern that trigger changes in capacitance when the rotor plate is in motion. All of these methods require manual assembly of the capacitive copper plates into a (3D printed) mechanism. In contrast, MechSense enables users to print the entire object with integrated mechanisms and rotational sensors in one pass.

2.2 3D Printed Sensors

More recently, researchers started to use 3D printing with conductive filament to fabricate objects with integrated sensors in one go. One example of this is the integration of direct touch sensing with 3D printed objects. *Capricate* [18] and *PrintPut* [1], for instance, provide editors to embed touch sensors, sliders and touch pads into 3D printable objects. Similarly, *Let's Frets!* [14] is a 3D printed capacitive fretboard that detects the user's fingers to help teach guitar playing. In addition to 3D printing the sensor geometry, *ModElec* [7] also automatically creates the internal conductive circuit traces necessary to route the sensor to an external microcontroller.

Rather than using 3D printing with conductive filament for direct touch sensing, researchers also investigated how to embed sensors that interact with other objects, such as capacitive surfaces. For instance, *Flexibles* [20] are tangible objects with embedded conductive material whose deformation can be sensed via a capacitive touch screen. *CAPath* [11] extends this work by also providing tangible sliders and knobs. *itsy-bits* [19] are 3D printed tangibles with embedded conductive markers that can be used to identify which tangible is used on a capacitive screen. Similarly, *3D-Auth* [13] are 3D printed tangibles with conductive patterns, which allow for two-factor authentication. *Off-line sensing* [17] 3D prints conductive sensors inside 3D objects connected to channels that contain liquids. When the object moves, the liquid bridges the gap between the capacitive areas, which can be used to detect orientation. However, none of these works investigate how to use conductive 3D printing to embed sensing into mechanical elements.

2.3 Integrating Sensing into Mechanical Elements

Over the last decade, the majority of work that added sensing to 3D printed mechanical elements used external sensors. For example, *Lamello* [16] uses acoustic sensing to detect interaction with physical sliders and rotary knobs, i.e. when the mechanism gets rotated or moved, it makes a noise which can be sensed with a microphone. *MechaMagnets* [24] integrates hall effect sensors and magnets into 3D printed objects for haptic and physical motion feedback. *StrutModeling* [12] uses rotational encoders in the metal connectors of 3D printed strut elements to sense how users assemble the struts. *3D Printing Wireless Connected Objects* [9] 3D prints gears and springs and adds conductive copper tape after printing has finished to create wifi signals when the mechanisms move.

More closely related to our work is research that uses conductive multi-material 3D printing to integrate sensing into mechanical structure. *MetaSense* [5] uses conductive 3D printing to integrate shear sensing into mechanical metamaterials. *FlexKeys* [6] similarly uses conductive multi-material 3D printing to create deformable springs that require no support material and can be used as sensors integrated with input devices such as keyboards. However, these works do not show how to integrate sensing with rotating mechanical elements, such as gears and linkages. While *SteelSense* [22] focuses on rotating mechanical elements, such as gears, hinges, screws, and bearings, it requires assembly since the conductive parts are metal-cast separately from the rest of the 3D print.

In summary, the existing work either requires multiple fabrication steps and manual assembly, or does not support rotating mechanisms, such as gears, linkages, and wheels. In contrast, *MechSense* provides an end-to-end fabrication pipeline for integrating *MechSense* encoders into 3D printable mechanism, provides a sensor layout that generalizes across different rotational mechanisms, and contributes a computational model to convert raw sensor data into angular position, direction of rotation, and rotational speed.

3 SURVEY OF HARDWARE DESIGN PROFESSIONALS

We conducted an exploratory survey of product design engineering professionals at a cross-industry consulting company with a small hardware design division to understand their familiarity with multi-material 3D printing, other smart materials, and the design engineering process in general. In doing so, we hoped to identify features that may make our design tool more effective, particularly in regard to existing utilized tools. 20 professionals responded to the survey including mechanical engineers (7), electrical engineers (7), industrial designers (5), and a product manager (1).

One key insight was the prevalence of participants who stated they had challenges integrating mechanical and electrical systems (50%). P17 explains 'mechanical constraints are sometimes in conflict with electrical requirements' and P13 goes on to say 'electronic component placement vs. mechanical volumetric constrains [are often a challenge].' Additionally, 14 participants shared interest in using or have already used advanced materials and manufacturing methods, such as multi-material 3D printing, with 18 being interested in creating smart products with these materials and methods in the future. Broadly, such responses indicated a need for better integration of mechanical and electrical components, and an opportunity to accelerate and simplify prototyping in electro-mechanical products.

In terms of existing design tools used by our respondents, we found SolidWorks is the most commonly used CAD modeling software (n=10), followed by Rhino (n=5) and Fusion360 (n=2). Based on these results and the prevalence of SolidWorks as a CAD tool in industry and academic settings more broadly with a market share of 22%(2021)[2], we created an add-on for SolidWorks that allows designers to automatically integrate the sensing directly into the design of the mechanical components for faster and easier prototyping. The demonstrated functionalities of our UI are agnostic of a particular CAD system and could also be implemented in other CAD systems.

4 MECHSENSE

MechSense is a method that uses multi-material 3D printing with conductive filament to augment 3D printed rotational mechanisms, such as gears, linkages, and shafts, with sensing capabilities. It

[2]https://www.cnccookbook.com/cnccookbook-2021-cad-survey-market-share-customer-satisfaction/

uses capacitive sensing to detect the speed of rotation, direction of rotation, and angular position.

4.1 Sensor Design

We developed a capacitive sensor design that incorporates a floating capacitor into the rotating component, such as the gear, and three sensor patches arranged in a circle around the static component, such as a shaft or a base plate (Figure 2) that can be fully 3D printed in one pass. Using capacitive sensing with this geometry eliminates the need for direct contact to the sensor patch in the moving component of the mechanism, thus allowing the moving component to rotate freely, providing a consistent sensor signal.

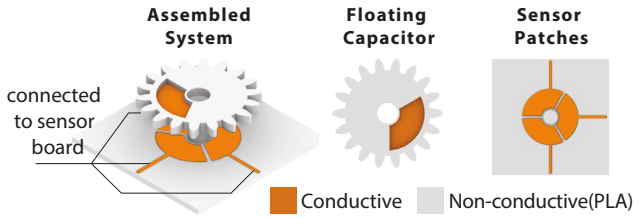

Figure 2: Our sensor layout consists of a floating capacitor in the moving part of the mechanism and three sensor patches on the static part of the mechanism that are wired to a capacitance sensing board.

The sensor patches on static part of the mechanism are connected to a capacitance sensing board (FDC2214 Sensor Board). The floating capacitor is integrated into the moving part of the mechanism and is not wired to the circuitry. Instead, the floating capacitor creates a coupled capacitance system between neighboring sensing patches. The amount of overlap between the floating capacitor and the sensor patches impacts the capacitance of the system, which we utilize to determine the location of the floating capacitor. This enables us to track the moving part of the mechanism while it is freely rotating.

4.2 Sensor Signal Properties

Figure 3 shows the signal of one sensor patch while the floating capacitor is rotating by 360°. We observe four different features in the signal: (1) a global minimum, (2+3) two maxima, and (4) a local minimum.

Global Minimum (No Overlap): When the floating capacitor and the active sensor patch have no overlap, we observe a *global minimum* in the signal (Figure 3 at 300° - 60°). The reason for this is that the floating capacitor is not coupled with the sensor patch and thus the detected capacitance is lower than in the other states.

Maxima #1 and #2 (Half Overlap): When the floating capacitor evenly overlaps with the active sensor patch and one of the neighboring patches, we observe two maxima at 120° and 240° (Figure 3). Since the neighboring patches are grounded by being connected to the sensing board, the coupled capacitance of the system is maximal in these positions. Although the shape of the sensing patches is identical, differences in print quality create different conductivity across patches which results in different peaks for each sensor.

Local Minimum (Full Overlap): When the floating capacitor and the active sensor patch are aligned, we observe a *local minimum*. The reason for this is that the floating capacitor has no overlap with the neighboring patches but is still in close proximity to them, leading to a small but measurable coupled capacitance (Figure 3 at 180°) that is still higher than the global minimum where the floating capacitor has no overlap with the sensing patch (Figure 3 at 360°).

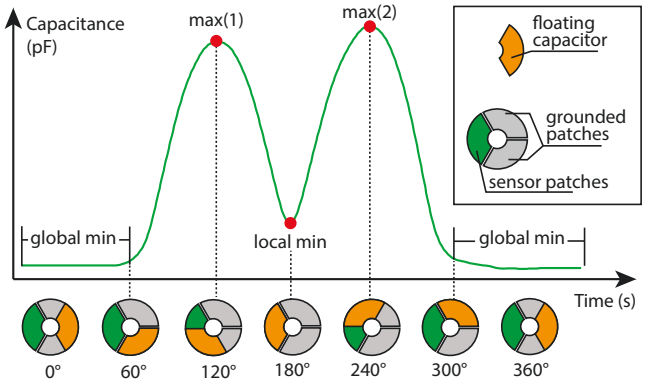

Figure 3: Signal profile of one sensor patch when the floating capacitor overlaps by different amounts and corresponding angular positions. We observe maxima and minima of the sensor signal at distinct positions of the floating capacitor.

4.3 Sensing Angular Position

Our goal is to derive the angular position from the capacitance values that we read from each of the sensor patches while the floating capacitor is rotating over them. To do that, we proceed with the following steps: (1) We preprocess the sensor data with a low-pass filter to reduce noise; (2) We determine the capacitance values of each extremum in the signal (e.g., the local minimum); (3) We divide the signal into 5 segments with each segment located between two extrema (Figure 4a); (4) We normalize the segments that are not in the global minimum and fit a polynomial to the sensor values that approximates the signal within each segment (Figure 4b). We can compute the angular position of the rotating element by determining the current signal segment it is in and estimating the current angle using the polynomial of this segment; and (5) we estimate the angle across all 3 sensor patches and calculate the average of all estimates. Since the sensor signal is constant in the global minimum across 120°(i.e., the floating capacitor has no overlap with the sensor patch), we cannot derive any angle in this region. Thus, we designed the sensor to contain 3 sensor patches such that the floating capacitor in the moving element always overlaps with at least one sensor patch at all times. If more than one sensor patch is not in the global minimum and outputs an angle estimate, we average between multiple sensor patch estimates.

Preprocessing of Raw Sensor Data: The raw sensor data contains noise that leads to the occurrence of multiple false local extrema.

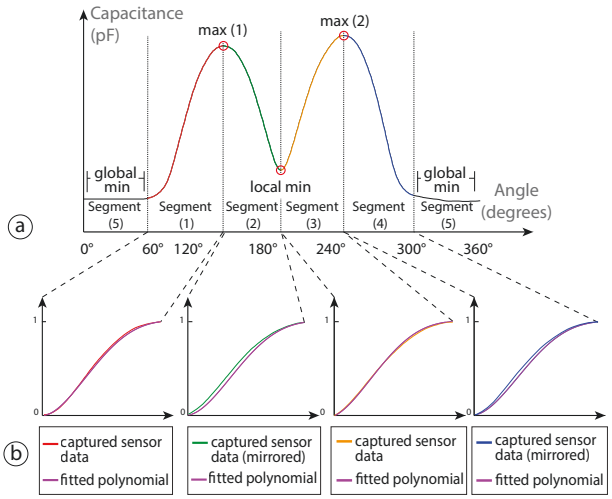

Figure 4: Processing of sensor data to estimate the angular position of the rotating element. (a) Segmenting the sensor signal into 5 segments. For each segment, there is only one possible angle for a sensorvalue. (b) Approximating segment 1-4 with a 4th-degree polynomial with segment 2 and 4 inverted such that all segments are monotonically increasing.

Hence, we apply a moving average filter to smooth the data. We chose a window size of 8 that is large enough to obtain a smooth signal but is also small enough to preserve the magnitude of the extrema.

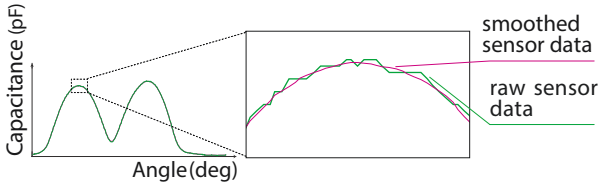

Figure 5: Smoothing sensor data. We apply a moving average filter (windows size = 8) to the raw sensor data to attenuate local fluctuations.

Detecting Extrema in the Sensor Signal: The key features for our angle estimation algorithm are the minima and maxima in the sensor signal over a full rotation. We find these extrema during an initial calibration step which has to be done once before using the sensor (Section 4.4). In this calibration step, we fully rotate the mechanism 3 times and use a peak detection method to identify the extrema. After storing the sensor values for these extrema, we use them for detecting extrema in live data by thresholding all incoming sensor values for these extrema points. We also take into account that the rotating element cannot randomly jump from one angular position to another, i.e. if the rotating element just passed through the local minimum, the next possible extremum can only be the maximum #2 if rotating clockwise (Figure 4a). Thus, we keep track of our last visited extrema and threshold only for the next possible extremum.

Segmenting the Sensor Signal: We divide the sensor signal into 5 segments where 4 segments are between two neighboring extrema of the signal and the fifth segment lies within the global mininum (Figure 4a). Within each segment, the sensor signal is strictly monotonically increasing or decreasing which means that for a specific sensing value there is only one possible angle position. We segment the sensor signal because one sensing value can have multiple possible angle positions throughout the entire signal.

Fitting a polynomial to each segment: Our goal is to derive a function that approximates all sensing values per segment and outputs an angular position for a sensor reading. To do that, we first ran an experiment by rotating a gear with an integrated MechSense encoder 50 times and captured the generated sensor data (3mm patch distance, $765mm^2$ sensor patch size, 10 RPM, Figure 19). We then segmented and normalized the sensor data as described above (Figure 4b). We fitted the data to several approximation functions including polynomials of degrees 3, 4, and 5 and a Fourier function. While all functions showed very similar approximation errors (<0.1°difference), we found that a 4th degree polynomial shows the least amount of error. Thus, we fitted a 4th degree polynomial to each of the 4 segments (that are not the constant global minimum), i.e. we generated 200 polynomials (50 rotations x 4 segments). Finally, we averaged the coefficients across all 200 polynomials. The resulting polynomial of the form $f(x) = ax^4 + bx^3 + cx^2 + dx + e$ has the coefficients $a = 1.74 * 10^{-7}$, $b = -2.86 * 10^{-5}$, $c = 0.00131$, $d = 0.0034$, and $e = -0.00863$.

After defining our approximation polynomial, we can now apply the polynomial directly to the normalized live sensing values of each segment which generates our angle estimation. We use the same polynomial for all angular position estimations across multiple prints and for all segments. Different sensor values at the extrema are compensated through the normalization step as well as thresholds for outliers, i.e. the magnitude of the minima and maxima might be different for an individual print which gets scaled to a range between 0 and 1 through the normalization step.

Detecting the current segment for Real-Time Sensing: When starting the sensor, the rotating element might be in an unknown position. To initially detect its position, we first identify the current signal segment by rotating through two extrema. While detecting only one extremum already allows us to detect its current position, it is necessary to go through two extrema in order to determine the current segment and thus, its direction of rotation.

In our implementation, we store the current location of the rotating element even when the sensor is unplugged. This allows us to keep track of the angular position also without an initial rotation through 2 extrema in case the floating capacitor has not been moved. In any other case, our algorithm can detect the angular position and the direction of rotation after passing through 2 extrema.

4.4 Calibration Phase

Before our computational model can estimate the angular position of the rotating element, we need to perform a calibration in which

we detect the sensing values at the extrema of the sensor signal, i.e., the local/global minima/maxima. To do that, we rotate the printed mechanism three times by 360° to collect sensor values for each feature (3 rotations x 4 features = 12 extrema values total). We detect the extrema by taking into account all 3 sensor signals of the patches. The direction of rotation is critical during the calibration step to detect the maxima in the right order. Thus, we require the calibration to be done only rotating clockwise or counterclockwise. Our software tool allows users to set the direction of rotation before starting the calibration.

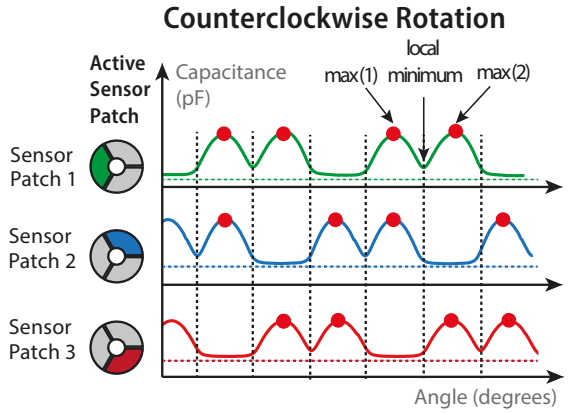

Figure 6: Detecting extrema during calibration using sensor data of multiple patches.

Global Minimum: We detect all sensor values that fall into the global minimum by finding the smallest and the largest sensing value throughout all rotations and define all values to be in the global minimum if they are within the lowest 20% percentile of sensor values.

Maximum #1: To find the value for the first maximum, we take into account the location of the global minima of the neighboring sensor patches. We detect the first maximum as the largest value that we read for sensor patch 1 while sensor patch 3 is in the global minimum (Figure 6). With the same strategy, we can detect the first maximum for patch 2 if patch 1 is in the global minimum, and for patch 3 if patch 2 is in the global minimum.

Maximum #2: Similarly to detecting the first maximum, we can detect the second maximum of sensor patch 1 by finding the largest sensing value of patch 1 while sensor patch 2 is in the global minimum. We can detect the second maximum of sensor patch 2, when the sensor value of patch 3 is in the global minimum, and the second maximum of patch 3, while sensor patch 1 is in the global minimum.

Local Minimum: To detect the local minimum we take into account the values that we found for the two maxima and find the minimum value that is between these two extrema.

4.5 Compensating for Sensing Value Variations at Extrema

It is possible that the sensor values at extrema change over time, e.g., due to a change in position or angle of the gear, humidity changes that influence the print material, or by the proximity of other capacitive objects like a human hand. To compensate for such variations we (1) modify the angular position estimation method to be robust against under- and overshooting of the senor values at extrema, and (2) we update the current estimation of the sensor values at extrema throughout multiple rotations.

Robustness of Extrema Detection: To compensate for overshooting at extrema points, we clamp the sensor values after normalization to cut out the outliers to 1 if they are larger than 1 and to 0 if they are smaller than 0. Once the sensor values surpass 1, we detect that as an extremum and switch from one segment to the next segment. It is also possible that sensor values undershoot, i.e., they never reach a value of 1. In this case, we take into account the sensor values of the neighboring patches. If one of the patches detects an extremum (i.e. the sensor values reach 1), we switch to the next segment globally even if the other sensor patches do not detect an extremum.

Updating for Extrema varations: To compensate for changes in the extrema values over multiple rotations we continuously update the extrema value using a moving average. For example, to update the sensor value for the local minimum, we identify the smallest value that we measured after passing through the local minimum. Next, we take the average between the old estimate of the local minimum and newly detected value, i.e., $LocalMin_{new} = 0.5 * (LocalMin_{old} + DetectedMin)$. We update all other extrema in the same manner.

4.6 Detecting Speed and Direction of Rotation

Speed: We compute the speed of rotation by measuring the change of angular position within a time frame. To do that, we attach a time stamp to the sensor data before sending it from the microcontroller to our implementation of the computational model and convert the change in angular position into Revolutions per Minute (RPM).

Direction of Rotation: To detect the direction of rotation, we first determine in which segment the sensor is currently. Each segment has monotonically increasing or decreasing values if rotated clockwise. For example, segment 1 only shows increasing values if rotated clockwise. In the case that the sensor outputs decreasing values while being in segment 1, we can identify a counterclockwise rotation. The same principle can be applied to all other segments with the exception of the global minimum which is nearly constant. Since only one sensor patch can be in the global minimum at a time, we can estimate the direction of rotation using the other two sensor patches.

5 APPLICATIONS

Our technique allows us to integrate sensing with various rotating mechanisms to enable a wide range of applications, such as a distance measuring wheel, a linkage based lamp, and a planetary gear box.

5.1 Distance Measuring Wheel

We integrated a MechSensor into a measurement wheel. If a user rolls the wheel over a surface, it can measure the distance traveled. We made this wheel by integrating 3 sensor patches ($765mm^2$ patch size, 3mm separation) and a floating capacitor into the wheel using our SolidWorks plugin. The entire device is fully 3D printed with the exception of the handle that we attach after the print. Next, we connected the sensor patches to our sensing board and added to our Processing UI a conversion from angle degree to cm by taking into account the diameter of the 3D printed wheel. Our software estimates the distance traveled with a mean error of 1.45mm according to the results of our technical evaluation 7. Since these wheels are often used for long-distance measurements, the tolerance can be acceptable for many use cases.

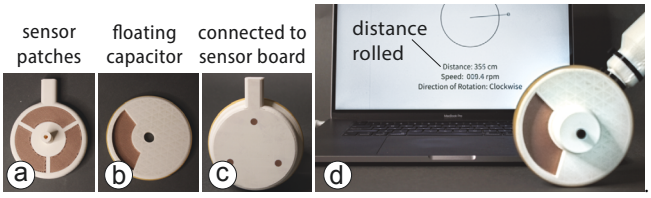

Figure 7: A distance measuring wheel with integrated sensing that can measure perimeters of irregular geometries and large open spaces.

5.2 Augmenting Everyday Objects

MechSense can also be used to augment everyday objects with smart behavior that adapts to the user's needs. We printed a smart desk lamp (Figure 8), which consists of two bars with linkages where the upper linkage contains a MechSensor ($76mm^2$ patch size with 3mm separation). When users want to look at something on a table more closely (e.g., when working with small objects like soldering SMD components), they can lower the lamp which our system detects and increases brightness. When the user wants to have regular ambient light again, the user can pull the lamp back to the standard position which lowers the brightness to the standard level.

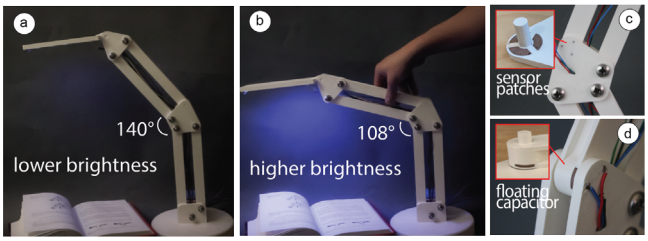

Figure 8: The smart desk lamp allows the user to change the brightness of the light by rotating the top linkage. To do so, we integrated a Mechsense encoder at the joint connecting the top and bottom linkages.

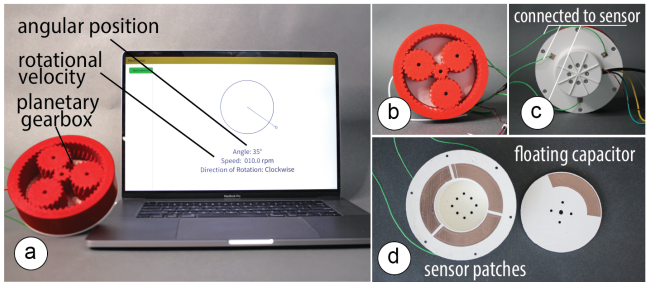

Figure 9: This planetary gearbox can sense its own angular position and rotational velocity.

5.3 Planetary Gear Box

Integrating sensors directly into a mechanisms geometry is particularly useful in space-constrained applications. One example of this is a planetary gearbox, which integrates multiple gears into a confined volume (Figure 9). Such gearboxes can be integrated into robotic arms and connected to a motor for actuation. By printing the static part of the sensors into the motor casing and the moving part into the geometry of the planetary gears, we are able to measure the angular position and the rotational velocity without increasing the overall size of the joint or requiring extra geometry for mounting an external encoder.

5.4 Fishing Rod Game Controller

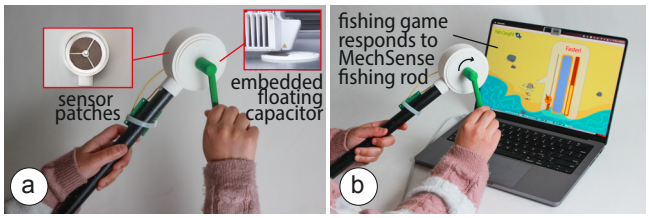

Figure 10: This fishing rod can sense its own angular position, rotational velocity, and number of rotations and can be used as a controller for gaming and VR experiences.

MechSense can also be utilized to fabricate rotary controllers for games and Virtual Reality (VR) experiences [21]. One such example is a fishing rod-shaped controller, which was printed in one pass with a MechSense encoder (Figure 10). In this example, the fishing rod interfaces with a 2D fishing game, where the user rotates the rod a certain number of turns to catch fish. The same controller can be used beyond 2D interfaces for VR to enhance experiences that hinge on rotational motion.

6 CREATING MECHSENSE OBJECTS

Our MechSense plugin for SolidWorks is written using C# to facilitate the creation of objects with integrated sensing. The plugin automatically integrates the sensor design into a CAD model of the mechanism and then enables exporting files for multi-material 3D printing. After 3D printing the mechanism, users need to wire the sensors to the microcontroller, and upload the sensing code which

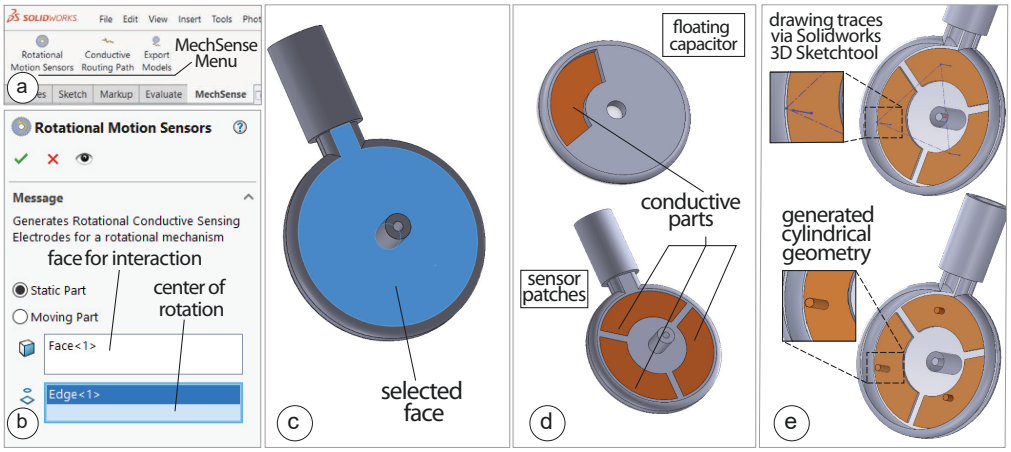

Figure 11: Integrating sensing into a mechanism using the MechSense 3D editor plugin.

streams raw sensor data to a computer connected via a serial port. We developed an implementation of our computational model and a UI for Java/Processing. The Processing UI enables users to calibrate the sensors. After finishing the calibration, our software converts the raw sensor data into the estimations for the angular position, direction of rotation, and the speed of the rotating element. The computational model is implemented as a Java library that can be imported to user-generated UIs and applications.

6.1 Designing the Mechanism

We developed a plugin for Solidworks that facilitates the integration of the sensor topology into a 3D mesh.

Integrating Sensor Patches into the Static Part: To integrate the sensor patches into the static part of the mechanism, the user selects the 'static part' option from the menu (Figure 11a). Next, the user selects the plane that should hold the sensor patches and the shaft that defines the center of rotation for the sensor patches. Given that information, our plugin generates the three sensor patches with 3mm distance between them, keeping a 1mm distance from the outlines of the static part geometry. Subsequently, our software creates separate meshes for 3D printing the sensor patches with conductive filament and the rest of the geometry with non-conductive filament.

Integrating the Floating Capacitor into the Moving Part: The user first selects the 'Moving Part' option from the menu bar (Figure 11). Next, the user selects the face onto which the floating capacitor should be integrated and the shaft of the mechanism that defines the center of rotation. Our MechSense plugin integrates the floating capacitor geometry with the moving part of the mechanism by subtracting it from the original geometry to generate separate files for the conductive and non-conductive parts for multi-material 3D printing.

Creating Traces: Users can integrate conductive traces to connect the sensor patches to the sensing board and microcontroller by using the built-in SolidWorks tool "3D Sketch". The user can draw lines for the conductive traces directly onto the geometry of the mechanism to connect the sensor patches to a connection point at a convenient location on the geometry. Our MechSense plugin converts this path into a cylindrical geometry of 2mm diameter for 3D printing upon the user selecting the "Conductive Routing Path" button.

Exporting Geometry and Sensing Code: On export, the MechSense editor separates the meshes for the conductive and non-conductive parts and generates separate .stl files for each of these components. The .stl files can be loaded into a slicing software for 3D printing where each file gets assigned a non-conductive and a conductive filament, respectively.

6.2 3D Printing

To manufacture the mechanism with integrated sensors in one pass, users load both non-conductive and conductive filament into a multi-material FDM 3D printer. Below, we provide more details on the conductive filament we used, the 3D printer hardware and print settings, as well as considerations regarding build plate adhesion.

Conductive Material: We use Electrifi filament from Multi3d[3] since it has the highest conductivity (0.006 Ω cm) among commercially available conductive thermoplastic filaments to date.

3D Printer Hardware and Print Settings: We use an Ultimaker S5 3D printer with a 0.6mm CC printcore from Ultimaker to accommodate the Electrifi conductive filament. The Electrifi filament is considerably softer than regular PLA filament and thus produces better print qualities with the use of a larger, abrasive material resistant printcore. To avoid grinding the softer conductive material during extrusion, we manually set the distance between the filament gears to the lowest feeder tension for our printer. In addition, to ensure that enough time is given to the conductive material to cool prior to the deposition of new layers, we set our print speed for Electrifi to 10 mm/s- 7mm/s, and non conductive PLA to 40 mm/s. The layer height was set to 0.15mm and wall thickness was set to 0.8mm. The print and build plate temperatures used were based on vendor recommendations.

[3]https://www.multi3dllc.com/product/electrifi/

Build Plate Adhesion: Electrifi filament does not adhere well to glass substrates such as the print platform of our 3D printer. Thus, we first print a layer of PLA on which the conductive traces can be reliably printed with sufficient adhesion. This first layer is automatically generated by our MechSense plugin for SolidWorks. (Figure 12).

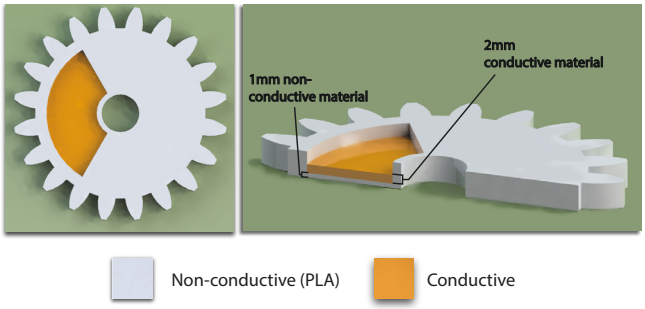

Figure 12: Build plate adhesion can be improved by printing a layer of PLA filament underneath the conductive filament.

Nozzle Print Speed: The Electrifi filament can smudge easily during printing. In particular, if two conductive areas, like the sensor patches, are close to each other, smudging can lead to short circuits between adjacent patches. This material behavior can be attributed to the low melting temperature of Electrifi, which can render it soft if not given enough time to cool down sufficiently. To alleviate print failures due to smudging, we reduced our print speeds from 15 mm/s to 7mm/s when printing the sensor patches, wires, and the floating capacitor to allow the conductive filament to cool down before a new layer is printed on top.

6.3 Connecting Sensors and Streaming Data

Users connect their 3D printed mechanisms to the sensing board and upload code to the microcontroller that streams the raw sensor values to the serial port.

Connecting Sensors to the Sensing Board: Since Electrifi has a high contact resistance, it is difficult to connect wires to printed traces just by taping them on. Instead, we integrate wires directly into the conductive material of the sensors by heating up the tip of the wires with a soldering iron and pushing them into the Electrifi traces. The hot tip melts the filament which allows us to push the wires in. This technique brings a large part of the wire's surface in contact with the conductive filament. After cooling down, the wires are tightly and reliably connected to the printed sensor.

Uploading Code and Retrieving Sensor Values: Users next upload our code to a microcontroller that is connected to a capacitive sensing board (FDC2214). Our code collects the read sensor values from the board using an open source library[4] to read data from the sensing board and streams them with a time stamp to our Processing UI via the serial port.

Sensing Board: We use a resonance-based capacitive sensor board that utilizes an RLC circuit (FDC2214, $50). This board has four sequential channels for capacitive sensing, with a capacitive sensing resolution of up to 28 bits (range: $1pF$ to $250nF$) at a sampling rate of 27ms.

[4]https://www.arduino.cc/reference/en/libraries/fdc2214

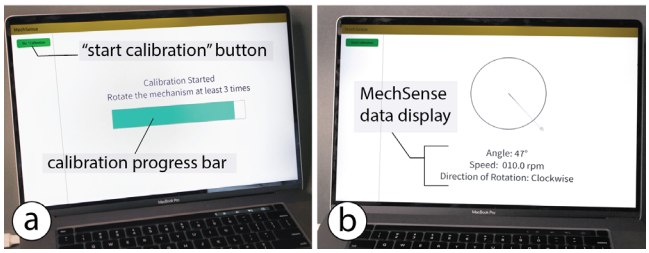

Figure 13: Calibrating and viewing MechSense sensor data through MechSense Sensor UI

6.4 MechSense UI to Process Raw Sensor Data

To make MechSense accessible to a wide range of users, we developed a Java/Processing implementation that supports users in performing the calibration step and converts the raw sensor data into angular position, direction of rotation, and rotational speed. To do the inital calibration, the user clicks on the button "Start Calibration". This loads a progress bar that asks users to do 3 rotations. When the calibration is completed, it reads the streamed data from the sensors and displays the angle, speed, and direction of rotation on the screen. The processing of the raw sensor data is implemented as a Java library that can be imported to any Java program and enables users to leverage MechSense for customized applications.

7 TECHNICAL EVALUATION

We ran a technical evaluation to determine the error in the angular position estimation for different spacing between sensor patches, the effect of different sensor patch sizes, and the influence of capacitive objects (such as the user's hand) on the sensing accuracy. We compared all angular position estimation to a commercial rotational magnetic encoder (14 bit encoder, AS5048) that acts as our ground truth.

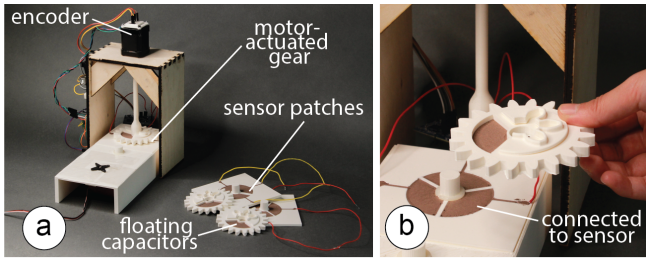

Figure 14: Evaluation Experiment Setup. (a) We connected a stepper motor with an integrated rotational encoder to a MechSense gear with an integrated floating capacitor. (b) The gear was mounted on a base plate with three sensor patches.

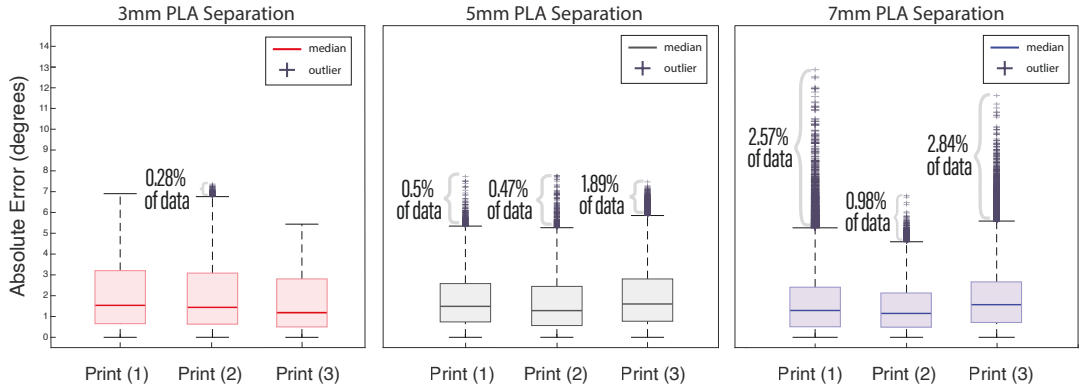

Figure 15: Error values retrieved from the angular position estimation for 50 rotations across 3 different prints for 3mm, 5mm, and 7mm PLA separation gaps between adjacent patches. The data demonstrates similar median error values (1.1°- 1.5°), but also shows an increase in the number of outlier values as the separation distance increases.

7.1 Spacing Between Sensors

To determine the influence of the separation distance between the sensor patches on the angular sensing accuracy, we conducted an experiment that evaluated the angular position estimations with sensor patches that have a separation of 3mm, 5mm, and 7mm.

Apparatus: We printed a contraption that can hold a removable base plate with three sensor patches that has a shaft in the middle to hold the gear with the floating capacitor. We added a washer between the plate and the gear, to ensure a constant distance of 1mm and minimize friction between the rotor-stator pair, and connected the gear to an axle that was mounted to a stepper motor (NEMA 17, 1.8° step size) (Figure 14a). We maintain a fixed minimum distance (2mm) between the floating capacitor and sensor patches (1mm washer thickness, 1mm PLA layer thickness). We considered only the minimal distance as it provides the maximum signal amplitude, and reduces the effect of environmental noise to the signal.

We evaluated 3 different spacings between sensor patches by printing 3 base plates with sensor patches separated by 3mm, 5mm, and 7mm (Figure 15) while keeping the total surface area constant at $765mm^2$. We also printed 3 matching gears whose floating capacitor area corresponded to the sensor patch area on the base plate. To capture variations in the sensing accuracy caused by the quality of the 3D print, we printed each plate-gear pair three times (3 conditions x 3 prints = 9 plate-gear pairs). For each plate-gear pair, we connected all three sensor patches to our sensing board and then placed each of the base plates and the matching rotating gears into the contraption.

We also attempted to print a plate with 1mm separation but noticed that, due to the printing resolution of our current 3D printer, the conductive material layers were not perfectly separated and created a short circuit between plates.

Procedure: We first calibrated each sensor by rotating the gear three times to generate the extrema estimations. After that, we captured the raw sensor data for each plate-gear pair for 50 rotations at a speed of 10 RPM with micro-stepping, and used our computational

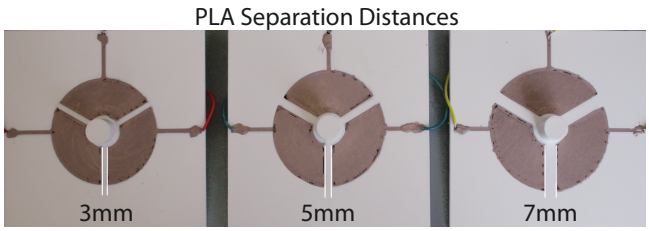

Figure 16: Separation distances between sensor patches: 3mm, 5mm, and 7mm.

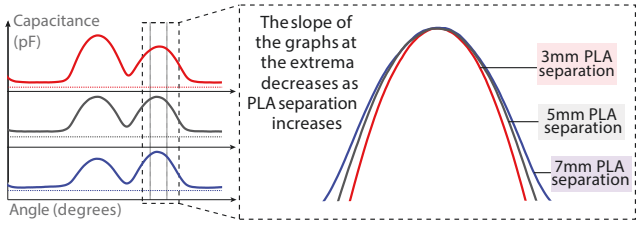

Figure 17: Comparison of the signal shape at a maximum for 3mm, 5mm, and 7mm PLA separation. We see a slight decreasing slope of the signal for increasing PLA separation distances.

model to convert the read data into the angular position estimation. Finally, we compared our estimate to the ground truth of the rotational encoder in the connected stepper motor and computed the error. **Results:** Figure 15 shows the results of the experiment.

All 9 plate-gear pairs show similar median errors between 1.1°- 1.5°. The 75^{th} Percentile of all captured error values reached up to 3% error across all experiment conditions, and the box plot whiskers (which are at the value of the 75^{th} Percentile + 1.5 * range between the 25^{th} and 75^{th} Percentile) reached up to 7° for the 3mm PLA separation, up to 5.9° for the 5mm PLA separation, and up to 5.5° for the 7mm PLA separation. We found that the majority of these larger

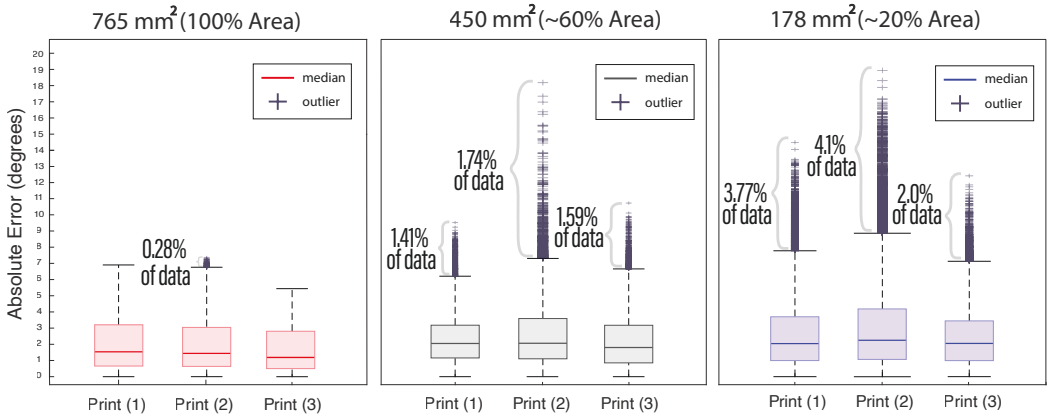

Figure 18: Angular position estimation error for sensor patches of $765mm^2$, $450mm^2$, $178mm^2$ surface area. We observe similar mean errors between all samples but an increase in outliers for smaller sensor sizes.

errors stem from misdetection of extrema near segment changes. In addition, we found an increasing amount of outliers for larger PLA separations. While the 3mm PLA separation had nearly no outliers, with one print having 0.28% of the captured data in the outlier range and an error of up to 7°, the 5mm PLA separation prints had a maximum of 1.89% of sensor values as outliers within a single print, and a maximum error of approximately 8°. The 7mm print's error value reached a maximum of around 13° for 2.57% of its sensor data as outliers. This increasing error can be explained by the change in the signal's shape for larger PLA separations. The distance between the plates leads to a wider shape of the sensor signal at maxima (Figure 17) which leads to errors in our polynomial for 5mm and 7mm patch distances since we trained it on a 3mm patch distance sample.

7.2 Effect of Sensor Size on Sensing Accuracy

To determine the influence of sensor patch size on the angular position estimation error, we conducted an experiment that evaluated the angular position estimation with sensor patches that had a surface area of $765mm^2$(100%), $450mm^2$(60%), and $178mm^2$(20%) (Figure 19).

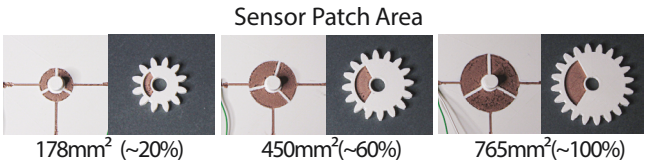

Figure 19: Evaluation with different sensor patches sizes.

Apparatus: We used the same experimental setup as in section 7.1, but this time we printed gears of different sizes with corresponding smaller sensor patch surface areas. We used a 3mm separation distance between sensor patches. We printed gears and base plates with matching floating capacitor and sensor patch areas three times for each combination (3 conditions x 3 prints = 9 plate-gear pairs in total). Since the $765mm^2$ sized gear with 3mm separation between sensor patches is identical to the experiment setup of section 7.1 for the 3mm separation case, we reused the captured sensor data in this experiment and compare it to the sensor patches with $450mm^2$, and $178mm^2$ patch surface area.

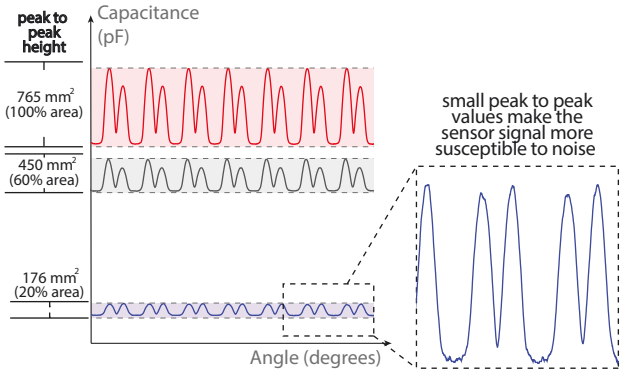

Figure 20: Sensor signal for patches with a total area of $765mm^2$, $450mm^2$, and $176mm^2$. Smaller sensor patches produce a smaller overall capacitance which makes them susceptible to external noise.

Procedure: For each plate-gear pair we conducted an initial calibration as described previously. Then, we rotated the gears at 10 RPM for 50 rotations each. We recorded the signal for each of the sensor patches and used our computational model to convert the raw sensor data into the angular position estimation. Finally, we used the rotational encoder in our stepper motor to generate the ground truth that we compare the sensor values to.

Results: Figure 18 shows the results of the experiment. We found that the mean error across all conditions was similar with 1.4°($765mm^2$), 2.0°($450mm^2$), and 2.1°($178mm^2$). However, we observed a growing amount of outliers for smaller sensor sizes, with 1.74% outliers

and a maximum error of 18.2° using the $450mm^2$ surface area.For $178mm^2$, we had 4.1% outliers with up to 18.9° error in the worst case. The increasing error for smaller sensor sizes can be explained by the change in signal strength of the captured data. In Figure 20, we can see that the amplitude of the signal decreases for smaller sensor sizes. This makes the sensor values at extrema less distinct, i.e., the change in capacitance becomes smaller, and may lead to misdetections at the extrema.

7.3 Effect of User's Hand Proximity on MechSense

We evaluated the angular position estimation error with a user's hand at 5 different distances from the MechSense encoder.

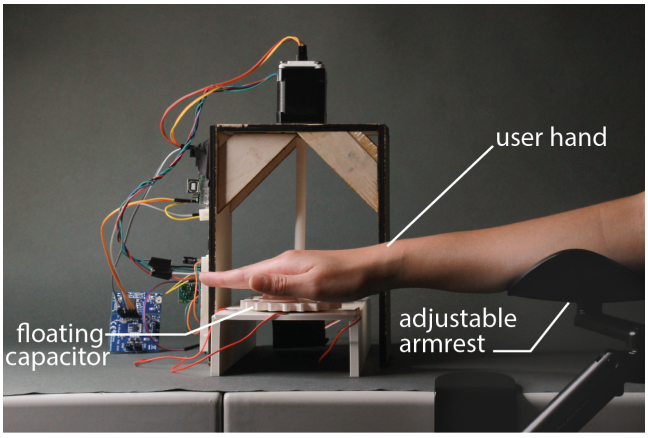

Figure 21: Experiment setup: user hand placed at different heights on top of rotating gear apparatus.

Apparatus: We used the same setup as in the previous evaluations but mounted a height-adjustable arm rest next to the MechSense encoder (Figure 21). We added the arm rest to provide the user a stable support to keep their hand at a constant distance from the MechSense encoder over a longer period of time. The sensor patches had an area of $765mm^2$ and a PLA separation of 3mm between each patch.

Procedure: We calibrated the sensor by rotating the gear three times without a user's hand in proximity. Next, we captured the angular position estimation error by rotating the gear 50 times at 10RPM with a user's hand at a fixed distance from the MechSense encoder. We asked a participant to place their hand on the arm rest which we set to a custom height and let them hover with their hand above the rotating gear. We repeated this procedure for 5 gear-hand distances: 50cm, 20cm, 10cm, 5cm, and 0cm (direct touch on the base plate). We measured the distance between the hand and the gear with a ruler and asked the participant to keep the hand as steady as possible. Each experiment took 5 minutes. In addition, we ran one experiment with no hand in proximity to the MechSense encoder to generate a baseline that allowed us to compare the error values with a hand in proximity.

Results: We observe a similar mean error of approximately 1.6° with the user's hand at a distance of 50cm, 20cm, and 10cm from the MechSense encoder. The amount of outliers also remains small with 0.064% (50cm), 0.042% (20cm), and 0.6% (10cm) of all sensor values with a maximum error of 15.6°(10cm). The mean error and amount of outliers increased when the user's hand is at a distance of 5cm or less (mean error= 2.1° (5cm) and 3.3° (direct touch)). The number of outliers increased to 3.24% (maximum error: 47.8°) of the sensor data for 5cm, and to 9.76% (maximum error: 58.3°) when directly touching the base with the sensor patches. This indicates that MechSense encoders experience increased angular position estimation errors in the presence of a user's hand below 10cm distance. Figure 23 shows the change in capacitance when a user's hand is present at 5cm distance. The capacitance values show a global increase that leads to detection errors of the extrema in our computational model. Our computational model updates the increased extrema values after a few rotations, thus, the majority of estimation errors stem from the first rotation after the hand approaches the sensor. However, this also indicates that the rapidly changing position of a user's hand will lead to further increases in angle detection errors as our computational model cannot adapt quickly enough.

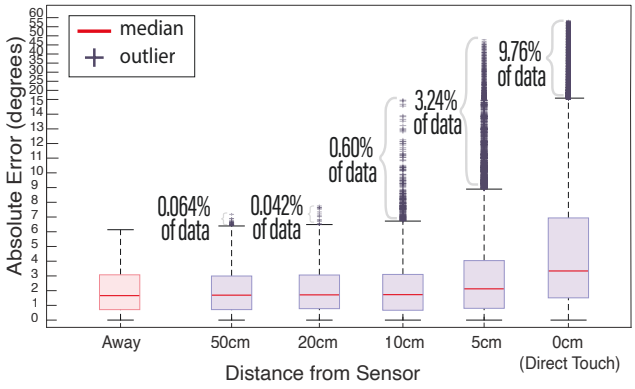

Figure 22: Angular position estimation error with a user's hand in 6 distances above the rotating gear: 50cm, 20cm, 10cm, 5cm, direct touch, and no hand in proximity.

8 LIMITATIONS AND FUTURE WORK

We next discuss the limitations of our work and potential avenues for future research.

Generating Conductive Traces: In our current user interface, the user has to manually route the traces, i.e., draw the path from the sensor patch to a location where they would like to connect the trace of the sensor patch to the sensing board. For future work, we plan to auto-route the traces.

Other Sensor Layouts: We also experimented with alternative sensor layouts (Figure 24). One sensor layout uses only two sensor patches (Figure 24a) requiring only two sensor channels on the sensing board, but it does not allow us to sense the direction of rotation since three sensor signals are required to eliminate ambiguity. The second layout (Figure 24b) used a double-sized floating

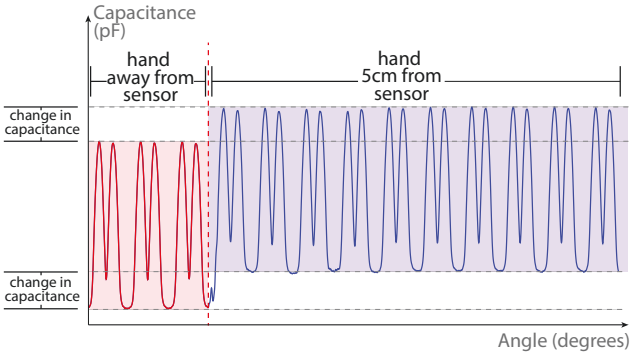

Figure 23: Capacitance change in the presence of a user's hand. We observe a global increase in capacitance in the presence of the user's hand (5cm distance).

capacitor. This produces a signal with three maxima corresponding to 25%, 50%, and 25% overlap as the floating capacitor moves across the sensor patch. For future work, we plan to further explore this sensor patch design to extract additional features based on slopes and intersections. Finally, we considered a gradient sensor layout (Figure 24c) in which the floating capacitor area increases. This allows us to extract the direction of rotation from one sensor patch only. However, we found that 3D printing the thin part of the gradient sensor patch was difficult to achieve but could be a promising method for larger mechanisms.

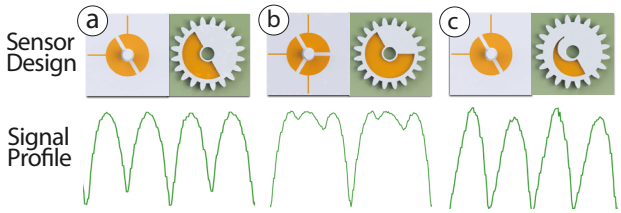

Figure 24: Alternative sensor layouts: (a) two sensor patches on the static part, (b) double-sized floating capacitor on the moving part, (c) gradient floating capacitor on the moving part.

Simulation of Sensor Signals: We plan to develop a computational prediction model to estimate the capacitance values of arbitrary sensor patch geometries and arrangements. This allows designing customized MechSensors that can be adapted to a specific prototype geometry. It thereby allows for rapid exploration of the parameter space, including the sensor patch shape and size and the dielectric strengths of the insulating material, which we could use in future work to enhance our computational prediction model for higher angular position accuracy.

Compensating for Proximity of a User's Hand: In future work, we want to compensate for noise induced by a user's hand by taking into account the change in capacitance for our global minimum. Since the increase in capacitance in proximity to a user's hand behaves similarly to a global lift-up of the original sensor signal on all three sensor patches, it might be possible to instantly remove these deviations by calculating the change at a global minimum and subtract this value from all sensor values. Since at least one sensor patch is in the global minimum at any time, this compensation could be computed almost instantly.

Sampling Rate & Speed Limitation: Our sensing board has a sampling rate of 27ms per sample which generates sensor data that approximates the true capacitance of the sensor at low rotational speed. However, at a higher rotational speed, the produced sensor data profile becomes less smooth with visible linear segments (Figure 25, 200RPM). One can see extrema are not perfectly captured at that rotational speed, which will introduce errors in our polynomial fitting and lead to an increasing sensing error. Application scenarios that require high rotation speed will have to utilize a sensing board that offers a higher sampling rate.

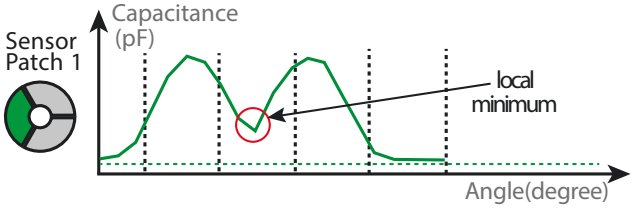

Figure 25: Sampling of the sensor signal at 200RPM at a sampling rate of 27ms. Fewer samples lead to a less smooth signal that results in larger angular estimation errors.

Infill Density & Pattern: In this work, we used vendor-recommended infill settings (100% infill density and line pattern) to maximize the conductivity of the sensing elements, and to eliminate any potential coupling that might occur due to gaps in the geometry. In the future, we plan to explore the effect of different infill settings on the capactive signal, and develop a computational model that maps these changes to the expected signal profile.

9 CONCLUSION

In this paper, we investigated how to integrate sensing into rotating mechanisms via conductive multi-material 3D printing to enable them to sense their direction of rotation, speed, and angular position. We showed how a sensor layout that integrates sensors with the static part of a mechanism and a floating capacitor with the moving part of the mechanism generalizes across different rotational mechanisms, such as gears, linkages, and wheels. We presented an editor that facilitates the integration of the sensors with the mechanism geometry, and that exports the 3D printable files. We also contribute a Java/Processing tool that uses our computational model to convert the raw sensor data into angular position estimation, direction and speed of rotation. We evaluated the angular position estimation error for different spacing between sensors, the size of the sensor patches, and the influence of the proximity of a user's hand near a MechSense encoder. For future work, we plan to explore how to increase the robustness of our sensing method to

external noise, and develop 3D printable sensors that can monitor other types of mechanisms.

ACKNOWLEDGMENTS

This project was supported by funding from Accenture LLP. We are grateful to Dr. Eric Gallo for his guidance and insightful discussions, and Taylor Tabb for his assistance with the design tool user study. We also thank the reviewers for their constructive feedback.

REFERENCES

[1] Jesse Burstyn, Nicholas Fellion, Paul Strohmeier, and Roel Vertegaal. 2015. PrintPut: Resistive and Capacitive Input Widgets for Interactive 3D Prints. In *Human-Computer Interaction – INTERACT 2015*, Julio Abascal, Simone Barbosa, Mirko Fetter, Tom Gross, Philippe Palanque, and Marco Winckler (Eds.). Springer International Publishing, Cham, 332–339.

[2] V. Ferrari, A. Ghisla, D. Marioli, and A. Taroni. 2004. Capacitive angular-position sensor with electrically-floating conductive rotor and measurement redundancy. In *Proceedings of the 21st IEEE Instrumentation and Measurement Technology Conference (IEEE Cat. No.04CH37510)*, Vol. 1. 195–200 Vol.1. https://doi.org/10.1109/IMTC.2004.1351027

[3] P.L. Fulmek, F. Wandling, W. Zdiarsky, G. Brasseur, and S.P. Cermak. 2002. Capacitive sensor for relative angle measurement. *IEEE Transactions on Instrumentation and Measurement* 51, 6 (2002), 1145–1149. https://doi.org/10.1109/TIM.2002.808052

[4] M. Gasulla, Xiujun Li, G.C.M. Meijer, L. van der Ham, and J.W. Spronck. 2002. A contactless capacitive angular-position sensor. In *SENSORS, 2002 IEEE*, Vol. 2. 880–884 vol.2. https://doi.org/10.1109/ICSENS.2002.1037224

[5] Jun Gong, Olivia Seow, Cedric Honnet, Jack Forman, and Stefanie Mueller. 2021. MetaSense: Integrating Sensing Capabilities into Mechanical Metamaterial. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 1063–1073. https://doi.org/10.1145/3472749.3474806

[6] Ben Greenspan, Eric M. Gallo, and Andreea Danielescu. 2022. FlexKeys: Rapidly Customizable 3D Printed Tactile Input Devices with No Assembly Required. In *Emilio Rossi and Massimo Di Nicolantonio (eds) Additive Manufacturing, Modeling Systems and 3D Prototyping (AHFE (2022))*. AHFE International, USA. https://doi.org/10.54941/ahfe1001587

[7] Liang He, Jarrid A. Wittkopf, Ji Won Jun, Kris Erickson, and Rafael Tico Ballagas. 2022. ModElec: A Design Tool for Prototyping Physical Computing Devices Using Conductive 3D Printing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 4, Article 159 (dec 2022), 20 pages. https://doi.org/10.1145/3495000

[8] Bo Hou, Zhang Tian, Cao Li, Qi Wei, Bin Zhou, and Rong Zhang. 2017. A capacitive rotary encoder with a novel sensitive electrode. In *2017 IEEE SENSORS*. 1–3. https://doi.org/10.1109/ICSENS.2017.8234143

[9] Vikram Iyer, Justin Chan, and Shyamnath Gollakota. 2017. 3D Printing Wireless Connected Objects. *ACM Trans. Graph.* 36, 6, Article 242 (nov 2017), 13 pages. https://doi.org/10.1145/3130800.3130822

[10] M. Karali, A. T. Karasahin, O. Keles, M. Kocak, and M. A. Erismis. 2018. A new capacitive rotary encoder based on analog synchronous demodulation. *Electrical Engineering* 100 (2018), 975–1983. https://doi.org/10.1007/s00202-018-0677-9

[11] Kunihiro Kato, Kaori Ikematsu, and Yoshihiro Kawahara. 2020. CAPath: 3D-Printed Interfaces with Conductive Points in Grid Layout to Extend Capacitive Touch Inputs. *Proc. ACM Hum.-Comput. Interact.* 4, ISS, Article 193 (Nov. 2020), 17 pages. https://doi.org/10.1145/3427321

[12] Danny Leen, Raf Ramakers, and Kris Luyten. 2017. StrutModeling: A Low-Fidelity Construction Kit to Iteratively Model, Test, and Adapt 3D Objects. In *Proceedings of the 30th Annual ACM Symposium on User Interface Software and Technology* (Québec City, QC, Canada) *(UIST '17)*. Association for Computing Machinery, New York, NY, USA, 471–479. https://doi.org/10.1145/3126594.3126643

[13] Karola Marky, Martin Schmitz, Verena Zimmermann, Martin Herbers, Kai Kunze, and Max Mühlhäuser. 2020. 3D-Auth: Two-Factor Authentication with Personalized 3D-Printed Items. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376189

[14] Karola Marky, Andreas Weiß, Florian Müller, Martin Schmitz, Max Mühlhäuser, and Thomas Kosch. 2021. Let's Frets! Mastering Guitar Playing with Capacitive Sensing and Visual Guidance. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI EA '21)*. Association for Computing Machinery, New York, NY, USA, Article 169, 4 pages. https://doi.org/10.1145/3411763.3451536

[15] Thijs Jan Roumen, Willi Müller, and Patrick Baudisch. 2018. Grafter: Remixing 3D-Printed Machines. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3173574.3173637

[16] Valkyrie Savage, Andrew Head, Björn Hartmann, Dan B. Goldman, Gautham Mysore, and Wilmot Li. 2015. Lamello: Passive Acoustic Sensing for Tangible Input Components. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. Association for Computing Machinery, New York, NY, USA, 1277–1280. https://doi.org/10.1145/2702123.2702207

[17] Martin Schmitz, Martin Herbers, Niloofar Dezfuli, Sebastian Günther, and Max Mühlhäuser. 2018. Off-Line Sensing: Memorizing Interactions in Passive 3D-Printed Objects. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–8. https://doi.org/10.1145/3173574.3173756

[18] Martin Schmitz, Mohammadreza Khalilbeigi, Matthias Balwierz, Roman Lissermann, Max Mühlhäuser, and Jürgen Steimle. 2015. Capricate: A Fabrication Pipeline to Design and 3D Print Capacitive Touch Sensors for Interactive Objects. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software & Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 253–258. https://doi.org/10.1145/2807442.2807503

[19] Martin Schmitz, Florian Müller, Max Mühlhäuser, Jan Riemann, and Huy Viet Viet Le. 2021. Itsy-Bits: Fabrication and Recognition of 3D-Printed Tangibles with Small Footprints on Capacitive Touchscreens. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 419, 12 pages. https://doi.org/10.1145/3411764.3445502

[20] Martin Schmitz, Jürgen Steimle, Jochen Huber, Niloofar Dezfuli, and Max Mühlhäuser. 2017. Flexibles: Deformation-Aware 3D-Printed Tangibles for Capacitive Touchscreens. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 1001–1014. https://doi.org/10.1145/3025453.3025663

[21] Dishita G. Turakhia, Harrison Mitchell Allen, Kayla DesPortes, and Stefanie Mueller. 2021. FabO: Integrating Fabrication with a Player's Gameplay in Existing Digital Games. In *Creativity and Cognition* (Virtual Event, Italy) *(C&C '21)*. Association for Computing Machinery, New York, NY, USA, Article 21, 10 pages. https://doi.org/10.1145/3450741.3465239

[22] Tatyana Vasilevitsky and Amit Zoran. 2016. Steel-Sense: Integrating Machine Elements with Sensors by Additive Manufacturing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 5731–5742. https://doi.org/10.1145/2858036.2858309

[23] Hewen Wang, Kai Peng, Xiaokang Liu, Zhicheng Yu, and Ziran Chen. 2021. Design and Realization of a Compact High-Precision Capacitive Absolute Angular Position Sensor Based on Time Grating. *IEEE Transactions on Industrial Electronics* 68, 4 (2021), 3548–3557. https://doi.org/10.1109/TIE.2020.2977540

[24] Clement Zheng, Jeeeun Kim, Daniel Leithinger, Mark D. Gross, and Ellen Yi-Luen Do. 2019. Mechamagnets: Designing and Fabricating Haptic and Functional Physical Inputs with Embedded Magnets. In *Proceedings of the Thirteenth International Conference on Tangible, Embedded, and Embodied Interaction* (Tempe, Arizona, USA) *(TEI '19)*. Association for Computing Machinery, New York, NY, USA, 325–334. https://doi.org/10.1145/3294109.3295622

[25] Dezhi Zheng, Shaobo Zhang, Shuai Wang, Chun Hu, and Xiaomeng Zhao. 2015. A Capacitive Rotary Encoder Based on Quadrature Modulation and Demodulation. *Instrumentation and Measurement, IEEE Transactions on* 64 (01 2015), 143–153. https://doi.org/10.1109/TIM.2014.2328456