# CapSense: A Real-Time Capacitive Sensor Simulation Framework for Physical Human-Robot Interaction

Christian Schöffmann, *Graduate Student Member, IEEE*, Zackory Erickson, *Member, IEEE*, and Hubert Zangl, *Member, IEEE*

***Abstract*—This letter presents CapSense, a real-time open-source capacitive sensor simulation framework for robotic applications. CapSense provides raw data of capacitive proximity sensors based on a fast and efficient 3D finite-element method (FEM) implementation. The proposed framework is interfaced to off-the-shelf robot and physics simulation environments to couple dynamic interaction of the environment with an electrostatic solver for capacitance computation in real-time. The FEM method proposed in this letter relies on a static tetrahedral mesh of the sensor surrounding without a-posteriori re-meshing and achieves high update rates by an adaptive update step. CapSense is flexible due to various configuration parameters (i.e. number, size, shape and location of electrodes) and serves as a platform for investigation of capacitive sensors in robotic applications. By using the proposed framework, researchers can simulate capacitive sensors in different scenarios and investigate these sensors and their configuration prior to installation and fabrication of real hardware. The proposed framework opens new research opportunities via sim-to-real transfer of capacitive sensing. The simulation approach is validated by comparing real-world results of different scenarios with simulation results. In order to showcase the benefits of CapSense in physical Human-Robot Interaction (pHRI), the framework is evaluated in a robotic healthcare scenario.**

***Index Terms*—Simulation and animation, physical human-robot interaction.**

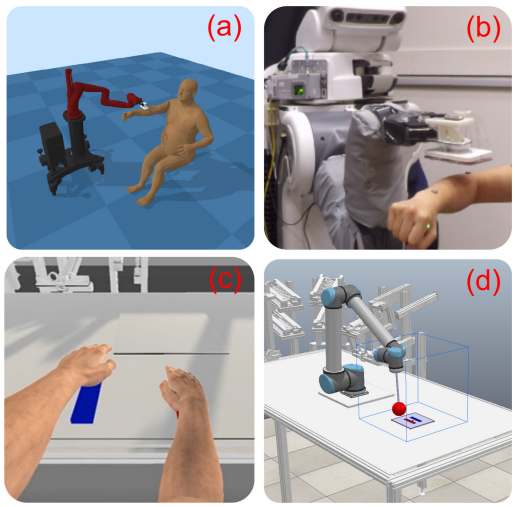


Fig. 1 Overview of different scenarios involving capacitive proximity perception and the proposed framework. Healthcare assistance scenario in simulation (a) and real-world scenario (b). (c) Sensing of human hands based on tracking input data and (d) object sensing in simulation.

## I. Introduction

PERCEPTION has become one of the major fields of research in robotics and one of the most important aspects for safe and reliable robotic systems. Vision-based sensor systems such as stereo vision, depth cameras or event-cameras gained high popularity among a variety of problems in robotics, yet these principles suffer from drawbacks. Especially in situations which are subject to dirt, moisture, fog or substantial occlusions, vision-based sensors show limitations. One potential sensing principle, which can overcome some of the limitations of vision-based approaches is capacitive sensing. The main properties of such sensors are the integration on different (flexible) surfaces, leading to development of large-scale robotic skins [1]. Furthermore, these sensors are robust against occlusions and mechanical impact, making them highly durable. Due to their physical principle, they can be implemented as proximity and tactile sensing units, leading to multi-modal approaches. Hence, they are potential candidates to be used complementary to vision-based sensors in future robotic applications. Simulation tools have become a key part in robotics research [2], in part due to the emergent topic of data-driven approaches such as deep reinforcement learning. These methods require massive amounts of training data which makes real-world training often infeasible and heavily time-consuming with real robotic hardware. Thus, simulation presents an opportunity to train large data-driven models for robots across

Manuscript received 24 February 2022; accepted 26 June 2022. Date of publication 19 July 2022; date of current version 29 July 2022. This letter was recommended for publication by Associate Editor H. Soh and Editor H. Kurniawati upon evaluation of the reviewers' comments. This work was supported in part by Kärntner Wirtschaftsförderung Fonds (KWF) and in part by European Regional Development Fund (EFRE) under Grant 26616/30969/44253 and Pattern-Skin under Grant 3520/34263749706. *(Corresponding author: Christian Schöffmann.)*

Christian Schöffmann is with the Institute of Smart System Technologies, University of Klagenfurt, 9020 Klagenfurt am Wörthersee, Austria (e-mail: christian.stetco@aau.at).

Zackory Erickson is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: zackory@cmu.edu).

Hubert Zangl is with the Institute of Smart System Technologies, University of Klagenfurt, 9020 Klagenfurt am Wörthersee, Austria, and also with the AAU SAL Ubiquitous Sensing Lab, 9020 Klagenfurt, Austria (e-mail: hubert.zangl@aau.at).

Digital Object Identifier 10.1109/LRA.2022.3191942

a number of contexts, which can be exceptionally challenging to accomplish on real-world robotic systems. Recently, methods to solve these problems were introduced, known as sim-to-real learning. Although most physics engines support RGB-D cameras, force/torque sensing and robotic kinematics [3]–[5], many other sensing modalities, such as capacitive sensing, have not been rigorously modeled in simulation [6]. Today, capacitive sensors are often simulated using off-the-shelf FEM solvers [7] that require extensive computational resources and are not available within most modern robotics physics engines. Providing a real-time capacitive sensor plug-in for robotic applications provides a number of benefits: i) researchers can easily integrate these sensors into their robotic environment and investigate control methodologies that use capacitive sensors, ii) having a customizable sensor configuration can support optimization of sensor parameters (e.g. electrode shape, sensor distribution) and iii) the simulated capacitance data can be used in robot learning frameworks and opens up new research directions for robot sensing methodologies. Fig. 1 shows potential applications and scenarios of the proposed simulation framework. The source code is available online (https://github.com/chstetco/capsense). Throughout this letter, the following contributions are made:

- Real-time simulation of raw capacitive signals for proximity sensing applications integrated into a modern physics-based simulation engine for robotics.
- Adjustable sensor configuration in simulation, leading to investigation of different sensor setups.
- Custom-made FEM implementation with highly optimized simulation step updates for moving objects or object which deform.
- Evaluation and validation of the proposed framework using real-world capacitive sensors for pHRI and assistive contexts.

The remainder of this letter is structured as follows: Section II gives an overview of related work. Section III gives a short overview of capacitive sensor fundamentals. The proposed simulation approach is described in Section IV. Section V describes the experimental setups used for evaluation and validation of the proposed framework. Finally, in Section VI a conclusion is given.

## II. Related Work

In recent years, there was increasing interest in using capacitive proximity sensors in the robotics community for various tasks. Due to their versatile integration on different surfaces, problems in collision avoidance using capacitive sensing technology [1], [8]–[10] were proposed and successfully implemented. Capacitive sensors have also been proposed for so-called pre-shaping in problems such as object grasping or manipulation [11], [12] and for haptic feedback [13]. Furthermore, in caregiving robotics, these sensors were successfully implemented for the task of dressing and bathing people using multi-dimensional capacitive sensors [14]–[16]. These contributions in the robotics community showed the promising use of this sensing technologies in addition to well-known devices such as cameras. Also, machine learning techniques using capacitive input data were recently proposed. The authors in [17] and [18] trained a neural network based on capacitive proximity data for recognition of different materials based on their dielectric properties.

With the advent of high-performance computing, simulation became a key tool in robotics for algorithmic development, especially combined with learning frameworks. Recently, simulation pipelines have been proposed which generate photo-realistic images, lidar and radar data [19]–[21]. Such simulation tools are promising candidates for sim-to-real learning methods in robotics. With regards to capacitive proximity sensor simulation, Lüttgen *et al.* [22] proposed a simulation method for capacitive touch sensors based on a combined simulation of circuits using SPICE tools and quasi-electrostatic simulation. In [23], an analytical formulation of circular capacitive sensors was developed, which led to a fast simulation model. The proposed method only applies to a limited amount of electrode topologies. The authors in [24] simulated a grid of capacitance sensors using a proximity detection sphere with radius of 15 cm and computed the closest point to the sensor on a human body model. This approach does not consider simulation of different material properties as well as electrode geometries and true field propagation. Also, a capacitive sensor model based on ray-type proximity sensors was proposed in [25] to acquire sensor readings embedded in a robotic gripper. The ray-type readings were interpolated to obtain an object region which is then fed into a 2D FEM program. Detailed capturing of object shapes as well as moving objects were not considered and the FEM solver was not optimized towards changing geometries. In addition to these specific implementations, also open-source software is available for simulating different capacitive sensors [26], [27]. However, these software tools focus specifically on the sensor design and optimization of sensor parameters and not on the usage of capacitive proximity sensors in applications.

## III. Theory

### A. Physical Background

The fundamental physical principle of capacitive sensors is based on propagation of electromagnetic fields in 3D space and can be fully described by Maxwell's equations. In practice, capacitive sensors are typically driven by low-frequency excitation signals in the kHz to low MHz range. Hence, the wavelength of the involved signal is large, leading to a quasi-static assumption of electric fields [28]. This leads to a reduced set of Maxwell's equations,

$$\nabla \times \mathbf{E}(x, y, z) = \mathbf{0} \tag{1}$$

$$\nabla \cdot \mathbf{E}(x, y, z) = \frac{\rho}{\varepsilon} \tag{2}$$

where $\mathbf{E}(x, y, z)$ is the electric field in 3D space, $\rho$ is the charge density and $\varepsilon$ is the permittivity of surrounding isotropic media, respectively. Since the quasi-static assumption holds, the electric field in every point in space can be computed by introducing a

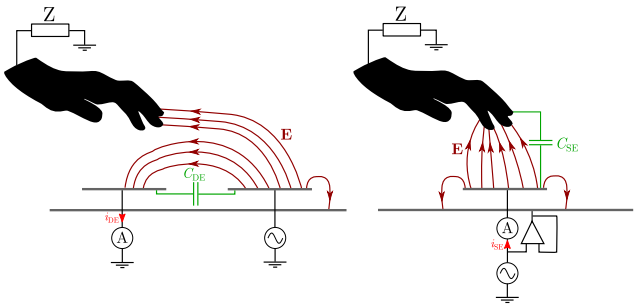


Fig. 2. Left: Mutual-capacitance sensing mode. Right: Self-capacitance sensing mode.

electric scalar potential $\phi(x, y, z)$ leading to

$$\mathbf{E}(x, y, z) = -\nabla\phi(x, y, z) \tag{3}$$

Inserting Eq. (2) into Eq. (3), one obtains

$$-\nabla \cdot (\varepsilon\nabla\phi(x, y, z)) = \rho \tag{4}$$

which is also known as the Poisson equation of electrostatics. In practice, capacitive sensing hardware measures the displacement current obtained from the displacement field

$$\mathbf{D}(x, y, z) = \varepsilon\mathbf{E}(x, y, z) \tag{5}$$

Integrating Eq. (5) over a surface area $\Gamma$ yields the surface charge within the integration area. Hence, in order to obtain the charge on an electrode of a capacitive sensor the charge must be computed. Inserting Eq. (3) into Eq. (5) and integrating gives

$$q = -\oint_{\Gamma} \varepsilon\nabla\phi(x, y, z)dr \tag{6}$$

Considering field lines normal to the electrode surface one obtains

$$q = -\oint_{\Gamma} \varepsilon\frac{\partial}{\partial\mathbf{n}}\phi(x, y, z)dr \tag{7}$$

where $\mathbf{n}$ is the inward normal of the boundary. Given the charge $q$, the capacitance $C$ can be computed straightforward from to the linear relationship between capacitance and charge, i.e. $C = q/V$.

### B. Finite Element Method

Eq. (4) is solved using numerical methods such as FEM. The solution to the FEM problem is obtained by subdividing the environment into mesh elements and approximating each element using interpolation functions such as piecewise linear functions of the form $\zeta_i(x, y, z) = r_0 + u_i x + v_i y + w_i z$ with coefficient vectors $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$, respectively. A so-called global stiffness matrix $\mathbf{K}^{\mathrm{G}}$ is assembled, which represents the system of linear equations to approximate the solution of the partial differential equation [29],

$$\mathbf{K}^{\mathrm{G}}\phi = \mathbf{b} \tag{8}$$

where $\phi$ is the vector of electric potentials for each node of the mesh and $\mathbf{b}$ is the so-called load vector, respectively. For the assembly of $\mathbf{K}^{\mathrm{G}}$, local stiffness matrices $\mathbf{K}_i^{\mathrm{L}}$ for each mesh

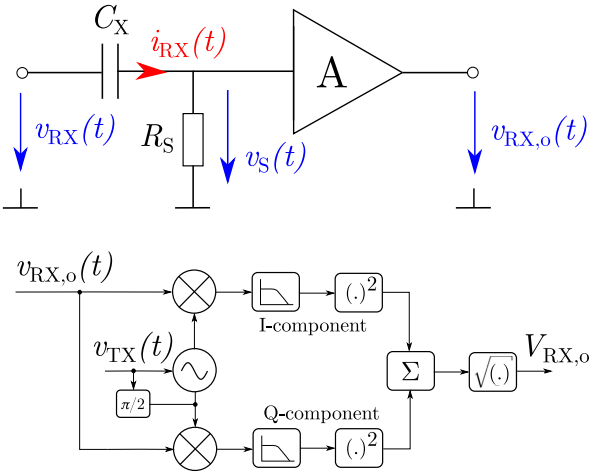


Fig. 3. Top: Schematic of the input stage to derive the capacitance of interest. Bottom: schematic of an IQ demodulation scheme as used by modern hardware.

element $n_i$ are computed,

$$\mathbf{K}_i^{\mathrm{L}} = \frac{\varepsilon_i}{36V_i}\mathbf{A}_i \tag{9}$$

where $\varepsilon_i$ is the permittivity of the $i^{\text{th}}$ mesh element, $V_i$ is the volume of the $i^{\text{th}}$ mesh element and $\mathbf{A}_i = \mathbf{u}\mathbf{u}^T + \mathbf{v}\mathbf{v}^T + \mathbf{w}\mathbf{w}^T$, respectively. The coefficient vectors are obtained from the coordinates of the corners of the respective mesh element. The global stiffness matrix is assembled from the local stiffness matrices, i.e. $\mathbf{K}^{\mathrm{G}} = \sum_i \mathbf{K}_i^{\mathrm{L}}$.

### C. Sensing Modes

Capacitive sensors are classified into two main realizations, i) self-capacitance mode and ii) mutual-capacitance mode, shown in Fig. 2. In mutual-capacitance mode, an electrode is excited by a harmonic signal, acting as a transmitter, whereas the second electrode is in receiving mode. An object in the vicinity of the receiver drains displacement current towards distant ground and blocks the electric field lines, leading to a change in capacitance $C_{\mathrm{DE}}$ between transmitter and receiver. In self-capacitance mode, only one electrode is driven and the self capacitance $C_{\mathrm{SE}}$ between the electrode and the object is sensed. Hence, the object acts as a counter electrode as opposed to mutual-capacitance mode where the object blocks field propagation. In practice, capacitive sensors consist of multiple electrodes to extract more information about the surrounding. Often, time-modulated excitation patterns during sensing are implemented, where a single electrode is set to transmitting mode whereas remaining electrodes are receiving (mutual-capacitance mode) or set to a bias value to be inactive (self-capacitance mode). Given $N_{\mathrm{E}}$ electrodes, the overall sensor response can be described by a matrix representation

$$\mathbf{C} = \begin{bmatrix} C_{1,1} & -C_{1,2} & \dots & -C_{1,N_{\mathrm{E}}} \\ -C_{2,1} & C_{2,2} & \dots & -C_{2,N_{\mathrm{E}}} \\ \vdots & \vdots & \ddots & \vdots \\ -C_{N_{\mathrm{E}},1} & \dots & \dots & C_{N_{\mathrm{E}},N_{\mathrm{E}}} \end{bmatrix}$$

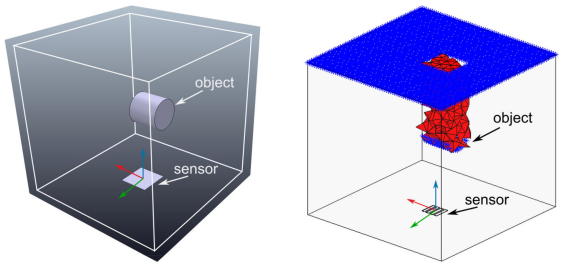


Fig. 4. The capacitive sensor is represented in the simulation environment by an orthographic camera. Based on the point-cloud information (blue points), mesh elements (red elements) are masked to represent the material distribution around the sensor.

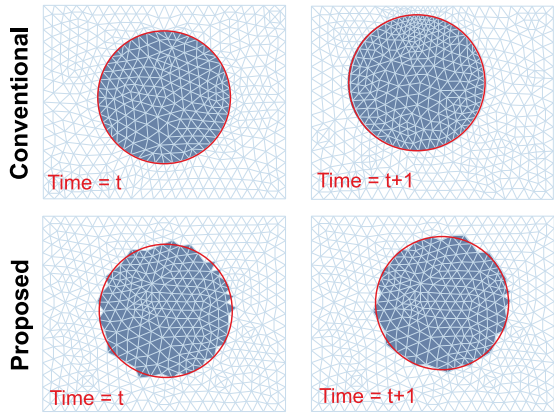


Fig. 5. Top row: Conventional FEM approaches use a moving mesh which updates with object movement, capturing the exact object outline. Bottom row: The proposed approach uses a fixed mesh.

The diagonal entries correspond to self-capacitance values $\mathbf{c}_{\text{SE}} = [C_{1,1} \ \dots \ C_{N_\text{E},N_\text{E}}]$ and the off-diagonal elements correspond to mutual-capacitance values $\mathbf{c}_{\text{DE}} = [C_{1,2} \ \dots \ C_{1,N_\text{E}}]$, respectively. Note that the terms self-capacitance and mutual-capacitance describe which capacitance coupling is measured. The physical quantity of interest is denoted as *capacitance*, which is used for both self and mutual-capacitance measurements in the following sections of the article.

### D. Measuring Hardware

From an electronics perspective, the receiver signal can be modeled as $v_{\text{RX}}(t) = V_{\text{RX}} \sin(\omega t + \varphi)$ where $\omega$ is the signal frequency and $\varphi$ the phase, respectively. Usually, the displacement current $i_{\text{RX}}(t)$ is measured as a voltage drop $v_{\text{S}}(t) = V_{\text{S}} \sin(\omega t + \varphi)$ at a known shunt resistance $R_S$ (see Fig. 3). Since the signal amplitude $V_{\text{S}}$ is very low, the measured voltage is amplified by an operational amplifier with gain $A$, yielding the amplified signal $v_{\text{RX,o}}(t) = V_{\text{RX,o}} \sin(\omega t + \varphi)$. The proximity information is encoded in the amplitude of $v_{\text{RX,o}}(t)$ and is retrieved using demodulation techniques such as IQ-demodulation (see Fig. 3). In such a scheme, $v_{\text{RX}}(t)$ is mixed by the known transmitter signal $v_{\text{TX}}(t)$, low-pass filtered and the magnitude from the IQ components extracted. Using AC signal analysis, the capacitance $C_{\text{RX}}$ can be calculated from the voltage $V_{\text{RX,o}}$ based on a simple RC network and using the voltage divider rule $C_{\text{RX}} = 1/(\omega R_S(\frac{AV_{\text{RX}}}{V_{\text{RX,o}}} - 1))$.

## IV. Simulation Framework

### A. Sensor Model

The capacitive sensor is represented by an orthographic camera in simulation, generating depth and RGB images of the scene in real-time based on standard z-buffering and rendering methods [30], as shown in Fig. 4. Since capacitive sensors do not have a cone-shaped field-of-view, orthographic cameras yield more realistic sensing coverage then perspective-type cameras. Most real-world capacitive sensors mounted to a robot are limited in sensing range by up to 0.5 m. Hence, the orthographic view of the camera is set to a cube with a volume of 0.5 m $\times$ 0.5 m $\times$ 0.5 m. The depth information is transformed into a point-cloud based on the intrinsic camera parameters which is further used as an input to the custom-made and accelerated FEM solver. For human tissue, the permittivity usually is well modeled and can be considered known and fixed. For other types of materials, the framework allows to encode the permittivity of objects in the red channel of the RGB stream of the camera model. Hence, different material distributions can be simulated by applying different red values to the individual objects. The electrode shapes are encoded based on Standard Tessellation Language (STL) files, which yields flexibility in simulating arbitrary number of electrodes and electrode topologies.

### B. Mesh Model

The mesh elements are defined as linear tetrahedral elements and are represented by a triplet of node, edge and triangle matrices $\mathcal{M} = \{\mathbf{P}, \mathbf{E}, \mathbf{T}\}$, respectively. Such a representation format is also used by open-source mesh tools, e.g. Gmsh [31] which can be used to generate mesh data. In contrast to common FEM software, no re-meshing procedure is applied during consecutive simulation steps. Conventional FEM solvers often assume prior knowledge of object shapes, as no knowledge about the sensor surrounding is available during simulation. Thus, in common FEM solvers, object movement is represented by a so-called moving mesh. After every movement, the mesh follows the movement by re-meshing the FEM environment accordingly, as shown in Fig. 5 in the top row. The re-meshing step also implies full re-computation of $\mathbf{K}^{\text{G}}$ which is time-consuming, especially when the environment is finely discretized. The proposed approach uses a fixed mesh instead, leading to tremendous speed-up by overcoming the time-consuming re-meshing operation. However, discretization errors occur as there is no prior knowledge of the object outline. Also, if the object displacement is below the mesh element size, the mesh entries do not update. However, this effect depends on the mesh granularity and the impact on the simulation results is minor compared to the gained speed-up.

### C. Initialisation Step

The global stiffness matrix $\mathbf{K}^{\text{G}}$ is pre-computed based on the mesh data, as described in Section III-B. In addition, for each mesh element $n_i$, the centroid $\mathbf{s}_i$ is computed, which is further used in the update step. A material vector $\varepsilon$ is generated,

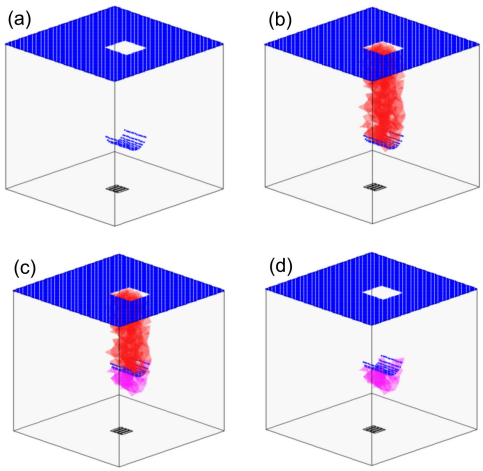


Fig. 6. (a) Point cloud information (blue points) is captured from the depth image. (b) Mesh elements (red elements) bounded by the point cloud are set to the material permittivity $\varepsilon_r$. (c) After object movement, the residual elements (magenta elements) are updated in $\mathbf{K}^G$ to the new material.

describing the electrical characteristics for each mesh element. At initialisation, it is assumed that the environment is free of any objects, i.e. $\varepsilon = \mathbf{1}$. To account for transmitting and receiving electrodes in the framework, Dirichlet boundary conditions need to be imposed on the electrode nodes. The solution of the electric potential at nodes belonging to transmitter and receiver electrodes is defined a-priori (e.g. imposed potential voltage or common ground potential). Since the solution at these nodes must not change, $\mathbf{K}^G$ must account for boundary conditions. Assume one transmitting electrode defined by a node vector $\mathbf{n}_{\text{TX}}$ set to a potential of $V_{\text{TX}} = 1$ V and one receiving electrode defined by a node vector $\mathbf{n}_{\text{RX}}$ set to ground potential, i.e. $V_{\text{RX}} = 0$ V. Since $[\phi_i]_{i\in\mathbf{n}_{\text{TX}}} = \mathbf{1}$ and $[\phi_i]_{i\in\mathbf{n}_{\text{RX}}} = \mathbf{0}$, one can manipulate $\mathbf{K}^G$ and $\mathbf{b}$ accordingly,

$$\begin{aligned}
&\mathbf{b}_j \longleftarrow \mathbf{b}_j - V_{\text{TX}}[\mathbf{K}^G]_{j,\mathbf{n}_{\text{TX}}} \quad \forall\, j \neq \mathbf{n}_{\text{TX}}\\
&\mathbf{b}_{\mathbf{n}_{\text{TX}}} = V_{\text{TX}}\\
&[\mathbf{K}^G]_{\mathbf{n}_{\text{RX}},\mathbf{n}_{\text{RX}}} = \mathbf{1}\\
&[\mathbf{K}^G]_{j,\mathbf{n}_{\text{TX}}} = [\mathbf{K}^G]_{\mathbf{n}_{\text{TX}},j} = \mathbf{0} \quad \forall\, j \neq \mathbf{n}_{\text{TX}}
\end{aligned}$$

For multiple electrodes, this procedure is applied for all electrode nodes, respectively.

### D. Update and Computation Step

After capturing the scene by the depth camera, mesh elements which are occupied by surrounding objects are masked based on the information from the depth image. The entries of the material vector $\varepsilon$ corresponding to these mesh elements are set to the material value $\varepsilon_r$ of the object, as illustrated in Fig. 6. It is important to note that the backside of the object is not directly visible to the sensor. Hence, remaining mesh elements at the backside are also set to the material $\varepsilon_r$. Since the boundaries of the sensor region are set to ground potential, the objects have low impedance towards ground. This is a valid assumption for human tissue and conductive objects. Moving objects in the vicinity of the sensor lead to variations of the electric field in 3D space. To account for such changes in the capacitive response, $\mathbf{K}^{\text{G}}$ must be updated according to the material distribution in the sensor's vicinity. Instead of updating the full global stiffness matrix, the proposed framework updates residual entries of the matrix. This yields speed-up in computation since only few mesh elements change between two consecutive simulation steps. The changed mesh elements are obtained by computing a residual material vector $\Delta\varepsilon_{(t)}$ at timestep $t$ which is further used for the matrix update, as shown in Fig. 6. The residual vector is obtained as $\Delta\varepsilon_t = \varepsilon_t - \varepsilon_{t-1}$ where $\boldsymbol{\varepsilon}_t$ and $\boldsymbol{\varepsilon}_{t-1}$ are the material vectors at time step $t$ and $t-1$, respectively. For the update, the non-zero elements are extracted and for each changed mesh element the local stiffness matrix is updated based on the differential change in permittivity

$$\mathbf{K}^L_{i,t} = \mathbf{K}^L_{i,t} + \Delta\varepsilon_{i,t}\mathbf{K}^L_{i,t-1} \tag{10}$$

Finally, a modified matrix $\tilde{\mathbf{K}}^G$ is obtained as described in Section III-B. The solution of nodal potentials is obtained by solving the linear system of equations using the modified global stiffness matrix

$$\tilde{\mathbf{K}}^G\phi = \mathbf{b} \tag{11}$$

After solving Eq. 11, the capacitances are directly obtained from the node potentials using the *unmodified* global stiffness matrix $\mathbf{K}^G$. The computation is performed without the need for gradient computation, i.e. $C_{i,j} = \frac{1}{u_j}\sum_{n_j}\mathbf{k}^T_{G,n_j}u_i$, where $\mathbf{k}^T_{G,n_j}$ are the rows of $\mathbf{K}^G$ corresponding to the receiving nodes $n_j$, $u_i$ and $u_j$ are the nodal electrical potentials of the transmitting and receiving electrodes, respectively. In case of self-capacitance sensing mode, the capacitance is obtained by setting $i = j$.

## V. Results

All computer experiments were conducted using an office computer with an Intel i7 processor rated at 3.40 GHz and 16 GB RAM. As sensing elements, custom-made capacitive sensors fabricated on a printed-circuit board (PCB) with FR4 substrate material were used. The simulation framework was tested and evaluated within the game engine Unity3D [32], the robot simulation platform CoppeliaSim [33] and the healthcare robot simulator Assistive Gym [34]. For the capacitive read-out hardware, a USRP X310 software-defined radio (SDR) platform with custom-made receiver daughterboards was used for mutual capacitance measurements [35]. For self-capacitance measurements a custom-made read-out hardware based on the AD7147 [36] Capacitance-to-Digital Converter (CDC) was used. The excitation frequency for all mutual capacitance experiments was set to $f_{\text{ex,DE}} = 1$ MHz whereas experiments conducted with the CDC were performed at $f_{\text{ex,SE}} = 250$ kHz. For all experiments, an offset calibration was performed to obtain the relative change of capacitance. The base capacitance was obtained as the capacitance reading without any objects in the vicinity of the sensor. The proposed framework was evaluated and validated on two scenarios involving capacitive sensors. First, a validation scenario with moving objects was implemented and three different electrode structures were investigated to highlight the flexibility of the proposed method. Finally, a healthcare robot

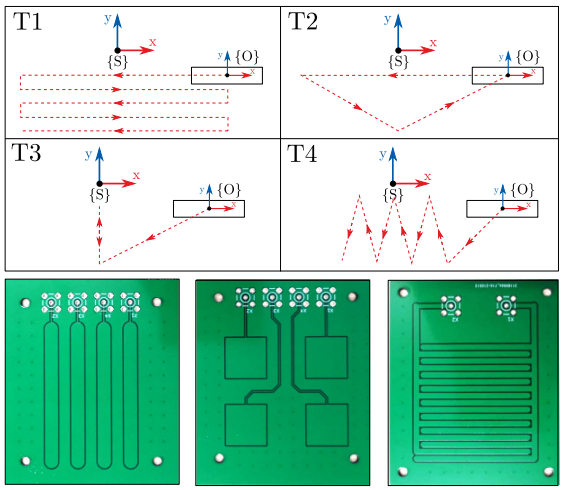


Fig. 7. Top: Trajectories implemented for validation of the proposed framework. Linear movement in x-direction (T1), triangular movement (T2), linear movement in y-direction (T3) and zig-zag movement (T4). $\{S\}$ denotes the sensor frame and $\{O\}$ is the initial start frame of the object. Bottom: Used electrodes for validation scenario.

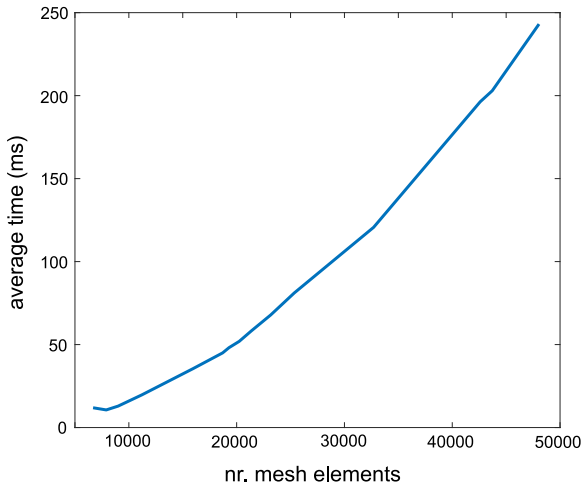


Fig. 8. Average timing behavior for the update and computation step for varying number of mesh elements.

scenario was investigated as a potential application scenario, where a robot performs human assistance tasks using capacitive sensors.

### A. Timing Analysis

To account for the performance of the proposed framework, timing analysis for the update and computation step described in Section IV-D was performed based on various mesh sizes. A sensor with three electrodes was simulated and the average computation time over 100 individual trials was computed for increasing mesh sizes. The object under investigation was a steel bar with dimensions of 150 mm × 150 mm × 20 mm moving on a pre-defined trajectory in the vicinity of the sensor. Overall, the simulation time scales nearly linearly with mesh elements, as shown in Fig. 8 and maintains average times below 100 ms for 30.000 mesh elements, which is sufficient granularity for pHRI scenarios.

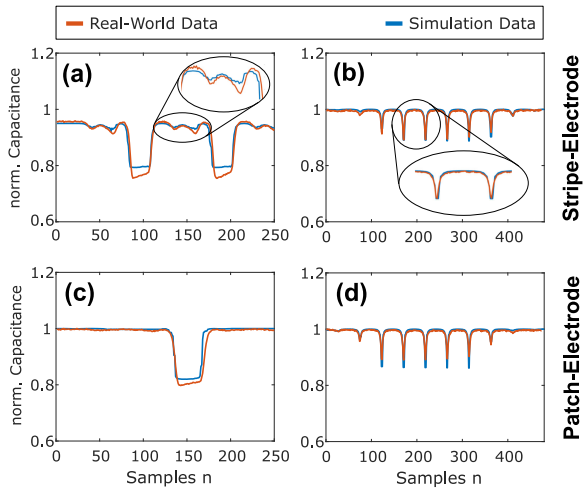


Fig. 9. Captured real-world and simulation data. (a) stripe electrode for triangular movement, (b) stripe electrode for zig-zag movement, (c) patch electrode for triangular movement and (d) patch electrode for zig-zag movement.

### B. Evaluation Scenario 1

For validation, three different electrode structures, as shown in Fig. 7, were used. The investigated electrodes are generic structures (stripes, patches, meander) often used for capacitive sensing. They allow to sense objects, gestures and to spatially resolve objects in space. The main purpose in using different electrodes is to highlight the flexibility of the proposed approach. The electrodes were mounted at a fixed position and a test object moved on pre-defined trajectories in the vicinity of the sensing electrodes. As a test object, a grounded steel bar with dimensions of 150 mm × 150 mm × 20 mm was used. In total, 4 different trajectories were implemented, as shown in Fig. 7. The object moved with two different speeds of $\vartheta_1 = 0.116$ m/s and $\vartheta_2 = 0.0166$ m/s. Fig. 9 shows example results of the proposed approach for two trajectories (triangular and zig-zag movement) and two different electrodes. The results show the sensor operating in mutual capacitance. Note that the simulation captures small variations (zoomed regions) in the response during movement and also different responses for the same trajectory due to the different electrode structures. For evaluation, the normalized cross-correlation coefficient was computed as a metric for signal similarity. The coefficient is given by

$$r_{c_\mathrm{s},c_\mathrm{r}} = \frac{\varphi_{c_\mathrm{s},c_\mathrm{r}}[0]}{\sqrt{\varphi_{c_\mathrm{s},c_\mathrm{s}}[n]\varphi_{c_\mathrm{r},c_\mathrm{r}}[n]}} \tag{12}$$

where $\varphi_{c_\mathrm{s},c_\mathrm{r}}[n]$ is defined as the cross-correlation between simulated signals $c_\mathrm{s}[n]$ and real-world signals $c_\mathrm{r}[n]$ given by $\varphi_{c_\mathrm{s},c_\mathrm{r}}[n] = \sum_n c_\mathrm{s}[n]c_\mathrm{r}[n+m]$. The signals $\varphi_{c_\mathrm{s},c_\mathrm{s}}[n]$ and $\varphi_{c_\mathrm{r},c_\mathrm{r}}[n]$ correspond to the auto-correlations of the simulated and real-world signals, respectively. Overall, the average correlation coefficient for the different electrode structures averaged over all four trajectories is summarized in Table I.

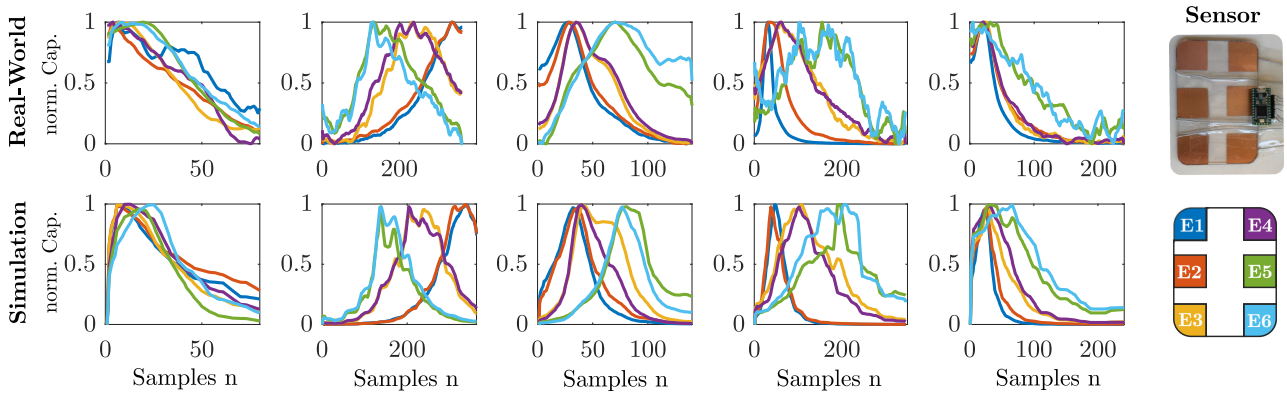


Fig. 10. Obtained results from real-world (top row) and simulated (bottom row) healthcare scenario as proposed in [16].

TABLE I
OBTAINED NORMALIZED CROSS-CORRELATION COEFFICIENTS FOR THE INVESTIGATED ELECTRODE TOPOLOGIES AVERAGED OVER ALL FOUR TRAJECTORIES

| | Stripe | Patch | Meander |
|---|---|---|---|
| Scenario 1, $v = 0.116$ m/s | 0.931 | 0.927 | 0.872 |
| Scenario 2, $v = 0.0166$ m/s | 0.947 | 0.953 | 0.905 |

TABLE II
OBTAINED NORMALIZED CROSS-CORRELATION COEFFICIENTS OF THE SIX ELECTRODES FOR THE HEALTHCARE EVALUATION SCENARIO AVERAGED OVER 20 RUNS PER PARTICIPANT

| | E1 | E2 | E3 | E4 | E5 | E6 |
|---|---|---|---|---|---|---|
| Participant 1 | 0.874 | 0.924 | 0.859 | 0.830 | 0.804 | 0.831 |
| Participant 2 | 0.881 | 0.895 | 0.856 | 0.911 | 0.892 | 0.878 |
| Participant 3 | 0.898 | 0.918 | 0.894 | 0.904 | 0.910 | 0.854 |
| Participant 4 | 0.908 | 0.918 | 0.893 | 0.908 | 0.881 | 0.882 |
| Participant 5 | 0.895 | 0.915 | 0.862 | 0.892 | 0.888 | 0.853 |
| Average | 0.891 | 0.914 | 0.872 | 0.889 | 0.875 | 0.859 |

### C. Evaluation Scenario 2

A potential application of capacitive proximity sensors is in pHRI. Especially in healthcare scenarios such as dressing or bathing of humans, capacitive proximity sensors show potential in control of robot arms near human limbs. Current systems rely on real-data collection processes which involve user studies and are therefore time-consuming. Overcoming this procedure using simulation-based evaluation and data collection can support new research directions and speed up progress in this area. The proposed framework was evaluated within the healthcare simulation environment Assistive Gym. Therefore, scenarios involving five different participants with 20 random trajectories per participant based from [16] were replicated in simulation and capacitance data near human limbs was collected. The human participants in simulation were modeled using the SMPL-X library [37] and robot control was implemented according to [16]. The capacitive sensor consists of six electrodes arranged in a 3-by-2 grid adhered to a polycarbonate plate. The whole sensor with the polycarbonate mounting plate is 115 mm $\times$ 85 mm $\times$ 1 mm in size and each electrode is a 30 mm $\times$ 30 mm square of copper foil driven in self-capacitance mode. Example data obtained by both real-world and simulated sensor is shown in Fig. 10. The results are in good accordance with the data obtained by the real sensor. It is worth mentioning that the results do not match perfectly, since ground truth retrieval of the exact human limb shape is not a trivial task and mesh discretization affects the precision of the simulation results. However, the human models of SMPL-X are very accurate, yielding to very similar sensor response in simulation. The capacitance data obtained from the simulation was normalized and compared to the real-world data using the metric described in Section V-B. The overall correlation coefficients for the six different electrodes of the evaluation scenarios are summarized in Table II.

## VI. CONCLUSION

this letter presented CapSense, a real-time, open-source simulation framework for capacitive proximity sensing in robotic applications. CapSense provides capacitive raw data in real-time based on a fast and efficient 3D FEM implementation. The raw data can be processed by arbitrary signal processing or machine learning algorithms, which potentially opens up new research directions in learning-based robotic control methodologies. The framework is fully configurable and allows simulation of different electrode configurations and topologies. The framework relies on a pre-defined mesh and an adaptive stiffness matrix update step, which yields simulation of moving or deformable objects in real-time without significant performance loss. The conducted experiments show the successful use of the simulation pipeline compared to real-world sensors in a lab environment and the validity of the data was investigated by means of a correlation metric. The framework is extendable by including soft-body physics for tactile sensing modalities and more detailed modelling of different materials. Also, approaches to model visual occlusions of conductive and non-conductive media using transparent shading techniques can extend the current work. However, it is believed that the presented simulation framework represents a major step towards realistic capacitive proximity sensor simulation frameworks for use in future robotic applications.

## REFERENCES


[1] K.-E. M' Colo, B. Luong, A. Crosnier, C. Néel, and P. Fraisse, "Obstacle avoidance using a capacitive skin for safe human-robot interaction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 6742–6747.

[2] H. Choi et al., "On the use of simulation in robotics: Opportunities challenges, and suggestions for moving forward," *PNAS*, vol. 118, no. 1, Jan. 2021, Paper e1907856118.

[3] A. Szot et al., "Habitat 2.0: Training home assistants to rearrange their habitat," *Adv. Neural Inform. Process. Syst.*, M. Ranzato, A. Y. Beygelzimer, P. Dauphin Liang, and J. W. Vaughan, Eds., vol. 34, pp. 251–266, 2021.

[4] M. Deitke et al., "Robothor: An open simulation-to-real embodied ai platform," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 3164–3174.

[5] B. Shen et al., "iGibson 1.0: A simulation environment for interactive tasks in large realistic scenes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021, pp. 7520–7527.

[6] M. M. Zhang, "Necessity for more realistic contact simulation," in *Robot.: Sci. Syst. (RSS) Workshop Visuotactile Sensors Robust Manipulation*, 2020.

[7] "AC/DC module user's guide," *COMSOL Multiphysics v. 6.0*, COMSOL AB, Stockholm, Sweden pp. 75–84, 2021.

[8] Y. Ding and U. Thomas, "Improving safety and accuracy of impedance controlled robot manipulators with proximity perception and proactive impact reactions," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2021, pp. 3816–3821.

[9] Y. Ding and U. Thomas, "Collision avoidance with proximity servoing for redundant serial robot manipulators," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2020, pp. 10249–10255.

[10] T. Schlegl, T. Kröger, A. Gaschler, O. Khatib, and H. Zangl, "Virtual whiskers – highly responsive robot collision avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2013, pp. 5373–5379.

[11] S. E. Navarro, S. Koch, and B. Hein, "3D contour following for a cylindrical end-effector using capacitive proximity sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2016, pp. 82–89.

[12] S. Escaida Navarro, M. Schonert, B. Hein, and H. Wörn, "6D proximity servoing for preshaping and haptic exploration using capacitive tactile proximity sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2014, pp. 7–14.

[13] H. Alagi, S. E. Navarro, J. Hergenhan, S. Musić, and B. Hein, "Teleoperation with tactile feedback based on a capacitive proximity sensor array," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, 2020, pp. 1–6, iSSN: 2642–2077.

[14] Z. Erickson, M. Collier, A. Kapusta, and C. C. Kemp, "Tracking human pose during robot-assisted dressing using single-axis capacitive proximity sensing," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 2245–2252, Jul. 2018.

[15] Z. Erickson, H. M. Clever, V. Gangaram, G. Turk, C. K. Liu, and C. C. Kemp, "Multidimensional capacitive sensing for robot-assisted dressing and bathing," in *Proc. IEEE 16th Int. Conf. Rehabil. Robot. (ICORR)*, 2019, pp. 224–231.

[16] Z. Erickson et al., "Characterizing multidimensional capacitive servoing for physical human-robot interaction," 2021, *arXiv:2105.11582*.

[17] H. Alagi, A. Heilig, S. E. Navarro, T. Kroeger, and B. Hein, "Material recognition using a capacitive proximity sensor with flexible spatial resolution," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 6284–6290.

[18] Y. Ding, H. Zhang, and U. Thomas, "Capacitive proximity sensor skin for contactless material detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2018, pp. 7179–7184.

[19] P. Martinez-Gonzalez, S. Oprea, A. Garcia-Garcia, A. Jover-Alvarez, S. Orts-Escolano, and J. Garcia-Rodriguez, "UnrealROX: An extremely photorealistic virtual reality environment for robotics simulations and synthetic data generation," *Virtual Reality*, 2019. [Online]. Available: https://doi.org/10.1007/s10055-019-00399-5

[20] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "FlightGoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2019, pp. 6941–6948. [Online]. Available: https://doi.org/10.1109/iros40897.2019.8968116

[21] C. Schöffmann, B. Ubezio, C. Böhm, S. Mühlbacher-Karrer, and H. Zangl, "Virtual radar: Real-time millimeter-wave radar sensor simulation for perception-driven robotics," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4704–4711, Jul. 2021.

[22] A. Lüttgen, S. K. Sharma, D. Zhou, D. Leigh, S. Sanders, and C. D. Sarris, "A fast simulation methodology for touch sensor panels: Formulation and experimental validation," *IEEE Sensors J.*, vol. 19, no. 3, pp. 996–1007, Feb. 2019.

[23] B. Huang, D. C. H. Yang, A. M. Madni, and J. B. Vuong, "Multielectrode differential capacitive sensors–model simulation with design application," *IEEE Sensors J.*, vol. 13, no. 4, pp. 1371–1379, Apr. 2013.

[24] A. Clegg, Z. Erickson, P. Grady, G. Turk, C. C. Kemp, and C. K. Liu, "Learning to collaborate from simulation for robot-assisted dressing," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 2746–2753, Apr. 2020.

[25] L.-M. Faller, C. Stetco, and H. Zangl, "Design of a novel gripper system with 3d- and inkjet-printed multimodal sensors for automated grasping of a forestry robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 5620–5627.

[26] M. S. Alnæs et al., "The fenics project version 1.5," *Arch. Numer. Softw.*, vol. 3, no. 100, 2015.

[27] F. Hecht, "New development in freefem," *J. Numer. Math.*, vol. 20, no. 3/4, pp. 251–265, 2012. [Online]. Available: https://freefem.org/

[28] J. D. Jackson, *Classical Electrodynamics*, 2nd ed. New York, NY, USA: Wiley, 1975. [Online]. Available: https://cds.cern.ch/record/100964

[29] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals*. 7th edition, Oxford, U.K.: Butterworth-Heinemann, 2013. [Online]. Available: https://doi.org/10.1016/B978-1-85617-633-0.00024-1

[30] T. Iler, E. Haines, and N. Hoffman, *Real-Time Rendering*, 4th Ed. Boca Raton, Fl, USA: CRC Press, 2018. [Online]. Available: https://books.google.at/books?id=0g1mDwAAQBAJ

[31] C. Geuzaine and J.-F. Remacle, "Gmsh: A. 3-d finite element mesh generator with built-in pre- and post-processing facilities," *Int. J. Numer. Methods Eng.*, vol. 79, no. 11, pp. 1309–1331, 2009. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.2579

[32] "Unity3d game engine," Unity Technologies, 2019, [Online]. Available: https://unity3d.com/

[33] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2013, pp. 1321–1326.

[34] Z. Erickson, V. Gangaram, A. Kapusta, C. K. Liu, and C. C. Kemp, "Assistive gym: A physics simulation framework for assistive robotics," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, 2020, pp. 10169–10176.

[35] L. Faller, T. Mitterer, J. P. Leitzke, and H. Zangl, "Design and evaluation of a fast, high-resolution sensor evaluation platform applied to mems position sensing," *IEEE Trans. Instrum. Meas.*, vol. 67, no. 5, pp. 1014–1027, May 2018.

[36] AD7147 CapTouch programmable controller for single-electrode capacitance sensors, analog devices, 2015, rev. E. [Online]. Available: https://www.analog.com/en/products/ad7147.html

[37] G. Pavlakos et al., "Expressive body capture: 3D hands, face, and body from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 10975–10985.