More Accurate Tactile Sensor Simulation with Hydroelastic Contacts in MuJoCo

David P. Leins[1]*, Florian Patzelt[1]* and Robert Haschke[1] .

Abstract— **This work presents a novel approach towards achieving highly realistic simulations of tactile sensors using MuJoCo in conjunction with hydroelastic contact surfaces, which replace standard point contacts. Hydroelastic contact surfaces compute a continuous pressure field on an extended contact surface, which allows us to implement tactile sensors with exceptional realism in the simulated environment. The extensions are implemented within a plugin framework, which we also used to realize ROS-control integration within MuJoCo, thus facilitating seamless transfer of robot control algorithms between simulation and real-world setups.**

Our experimental results demonstrate a remarkable resemblance between the sensor values obtained in the simulation and the data acquired from the physical sensors. Our approach's successful validation further substantiates our sensor simulation's efficacy in facilitating reliable sim-to-real transfer.

I. INTRODUCTION

Robot simulators have revolutionized robotics research and development by providing a cost-effective and secure environment for testing and refining robotic systems before real-world deployment. Traditionally, simulators have employed rigid object assumptions and point contacts to model interactions of any kind, for instance, robotic grippers and grasped objects. However, recent studies have exposed the limitations of these approaches, revealing non-physical artifacts and discrepancies when compared to real-world scenarios [1].

In an effort to address these limitations, MuJoCo [2] introduced soft-body point contacts, which improved upon the classical point contact model but still encountered issues due to rapidly changing contact point locations and force vectors, eventually leading to instabilities.

It is essential to drop the point-contact assumption and consider a continuous contact surface resulting from the deformation of objects to achieve greater realism in contact simulations. This deformation yields a non-uniform pressure distribution as well as smoothly evolving normal and frictional forces.

To this end, Elandt et al. [3] proposed hydroelastic contact surfaces as an approximate model to predict contact surfaces, pressure distributions, and the net contact wrench. This innovative approach combines the soft-body assumption and hydrostatic pressure, computing continuous pressure fields within the objects. The contact surface is defined at the equilibrium of overlapping pressure fields. Hydroelastic contact surfaces produce continuous wrenches, even for non-convex objects and coarse meshes, by incorporating dissipative, rate-dependent pressure and friction. This model exhibits faster evaluations than finite-element approaches while accurately capturing force, moment, and stiffness trends with contact load.

In this work, we leverage this approach originally developed for the Drake simulation [4] and provide an implementation integrated into MuJoCo. Additionally, we aim for a high-fidelity simulation of a barometric tactile sensor array [5], allowing for a smooth pressure representation for each sensor cell, further referenced as taxel, closely approximating the real-world sensor behavior.

The implementation of hydroelastic contact surfaces, along with the simulation of the barometric sensor array, is seamlessly integrated into our MuJoCo-ROS framework. This framework provides an easy-to-use interface for plugin integration and facilitates rapid sim-to-real transfer through its ROS interface, mirroring a real robotic system.

II. METHODS

A. *MuJoCo-ROS framework*

In this section, we present the MuJoCo-ROS framework, which serves a twofold purpose: (a) integrating a ROS interface into MuJoCo and (b) providing a versatile platform for incorporating new functionality into the simulator through plugins.

In robotics research, ROS has become a prevalent interface utilized by numerous robotic systems. We establish a unified interface that fosters testing scenarios under more realistic circumstances by establishing a direct connection between ROS and the simulation. This integration significantly facilitates the transfer of simulation results to real-world applications, promoting enhanced robot performance and efficiency.

A core feature of our framework is the provision of a high-level plugin system, allowing for easy extension of the simulation's capabilities. With plugins, researchers can implement new robots and sensors or even modify contact models, as exemplified by the plugin for hydroelastic contact surfaces. These plugins enable the simulation to be tailored to specific research needs, promoting the study of novel concepts and methodologies.

Plugins can override a set of predefined callbacks that are called at different stages of the MuJoCo's forward pass to interact with the simulation environment. Within these callbacks, plugins have full read and write access to the simulated world state at a given point in time, e.g., to apply

* equal contribution

[1]D. P. Leins, F. Patzelt and R. Haschke are with CITEC, Faculty of Technology, Bielefeld University, Germany. email: [dleins|fpatzelt]@techfak.uni-bielefeld.de.

This work was supported by the BMBF project Sim4Dexterity.

forces dynamically based on the current world state or to make simulation data available over ROS topics.

In summary, the MuJoCo-ROS framework serves as a valuable tool in robotics research, providing a bi-directional interface between ROS and the simulation environment. The incorporation of plugins allows researchers to easily extend and customize the simulator's functionalities, paving the way for innovative investigations while maintaining comprehensive sim-to-real transfer capabilities.

B. Hydroelastic Contact Surfaces

In order to compute the hydroelastic contact surfaces, it is necessary to obtain pressure fields for each object involved in the simulation. These pressure fields can be precomputed once at the start of the simulation using Laplace's equation, commonly employed in steady-state heat flow problems. For a given arbitrary object, the outer boundary has a strain of zero, while an internal boundary is set with a strain of 100%. Solving Laplace's equation with these boundary conditions yields a smooth field, approximating the strain value throughout the object. The strain field is then scaled by the elastic modulus, representing the material stiffness, to obtain the corresponding pressure field. The pressure field is approximated using a coarse tetrahedral volume mesh, ensuring rapid computations during the simulation process.

When two objects come into contact and penetrate each other, the hydroelastic contact surface is defined as the intersection area where the pressure values from both objects balance each other. This contact surface is represented by a surface mesh formed from the intersection of the pressure field meshes of the colliding objects. Consequently, the pressure and pressure gradient along the contact normal can be readily computed for each point on the contact surface.

The integral of pressure effects over the contact surface is calculated to determine the net contact wrench between the colliding objects. For a comprehensive understanding of the precise computation method, we refer to [3] and [6].

C. Tactile Sensors

In the real world, we use enhanced BMP388 digital barometer chips (with enlarged orifices) for tactile sensing [7]. These sensors are very well suited for a sim-to-real comparison baseline, as they display barely any hysteresis, making the integration of complex disturbances patterns in simulation obsolete.

In our lab, we use a 5x5 grid of those tactile sensor cells cast in silicone rubber with an inter-cell spacing of 2.2 mm [5] (see Fig. 1).

In simulation, the barometric sensor array is modeled as a compliant box, where the top surface corresponds to the silicone surface of the real sensor. Taxel locations, matching the cell positions of the real sensor, are defined on this surface. When the sensor comes into contact with another object, the output of the sensor is measured by first computing the hydroelastic contact surfaces and then approximating the pressure measured by each taxel by sampling points on the surface that fall into the taxel's receptive field and eventually aggregating their pressure values.

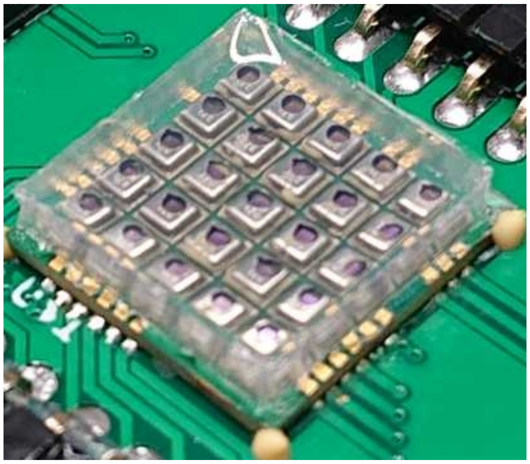

Fig. 1. 5x5 array of the enhanced BMP388 digital barometer chips cast in silicone rubber.

We pose the problem of sampling pressure along the contact surface as a rendering problem by interpreting taxels as pixels that should render the pressure values of the first intersection point between their normal direction and a surface mesh. The computer graphics research field, where this is a well-studied problem, offers the Möller-Trumbore algorithm [8] as one of the fastest solutions to the ray-triangle intersection problem. Further, we reduce the number of intersection tests to perform by employing another prominent acceleration strategy in computer graphics: bounding volume hierarchies (BVH), specifically Wald's binned building approach [9]. For each contact surface on a sensor, we construct a bottom-level acceleration structure (BLAS) based on axis-aligned bounding boxes and combine them in top-level acceleration structures (TLAS). These methods greatly reduce the number of time-consuming intersection tests, making the simulation of reasonably sized sensors feasible.

Since one ray per taxel will normally not give a good enough estimate for pressure within the receptive field, we introduce an additional resolution parameter to set the number of rays that are distributed uniformly within a taxel's receptive field. With this parameter, the trade-off between sensor receptiveness and computational cost can be tuned to the resources at hand and the use cases' necessities. The raycasting procedure is visualized in Fig. 2.

Multiple aggregation functions seem reasonable to compute the pressure output at each taxel from the samples. We employ a convex weighting that decreases quadratically with the distance from the taxel center:

$$p = \frac{\sum_{i=0}^{n-1} w_i p_i}{\sum_{i=0}^{n-1} w_i} \qquad w_i = (r - d_i)^2$$

n is the number of rays cast, p_i is the pressure at the point where ray i intersects the contact surface, r is the radius of the receptive field of the taxel, and d_i is the distance between the taxel sensor and ray i on the sensor surface.

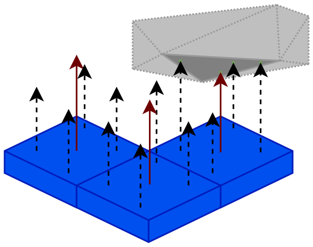

Fig. 2. Concept of raycasting-based taxels: the blue boxes represent individual taxels. The red arrows denote the main ray of each taxel, while the black arrows depict sub-samples.

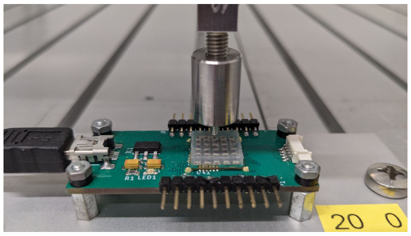

Fig. 3. Real world setup. The PCB carrying the sensor array is screwed onto the measuring table to keep the sensor fixed during measurements. Directly above the sensor array, the probe tip and part of the attached spring scale can be seen.

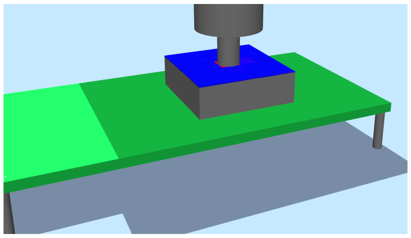

Fig. 4. MuJoCo-ROS simulated setup. The gray box on top of the green rectangle (representing the PCB) models the silicone block. The blue area on top of the silicone is the optional visualization of sensor cell activity. Based on the normalized pressure detected by the array, patches turn red upon detected contact.

III. EXPERIMENT

To validate our simulation approach, we conducted a set of measurements with the real sensor and its simulated counterpart.

A. Real Setup

In the real world, we used a 3-axis stepper motor setup carrying a cylindrical probe tip (⌀ 2 mm) attached to a spring scale. The three axes are controlled via an Arduino board running a grbl server. The digital interface of the spring scale provides the reference measurement of the force acting on the probe tip. The stepper motors allow setting the 3D position of the measure with a precision up to 0.0125 mm.

The barometric sensor array was centered within the measuring area and fixated with three screws (see Fig. 3). The probe tip traversed the x-axis of the sensor while centered on its y-axis, starting from the center of the leftmost sensor cell and stopping on the center of the rightmost cell with a step size of 0.0125 mm. For each x-axis position, the probe tip started from a safe height, i.e., not being in contact with the sensor, and moved down with steps of 0.0125 mm until a reference force of 0.35 N was measured. If the reference force was larger than 0.0 N, the current position, reference force, and barometric sensor value were saved. Due to the symmetry, a measurement on the x-axis is mostly identical to a measurement on the y-axis. For this reason, it suffices to only compare measurements on one axis.

B. Simulated Setup

In simulation, we modeled the 3-axis stepper motor with three very stiff joints to accurately mirror the motor precision. The target position is computed by a Python script and commanded via a ROS service call. To model the height displacement caused by a contraction of the spring scale, we employ a soft constraint acting as a spring-damper and increasing the applied force with a growing distance between the commanded and current tip position. The probe tip is modeled as a rigid cylinder, and the sensor as a soft box. The top surface of the box corresponds to the sensor surface. The simulated setup is visualized in Fig. 4. The measuring procedure is exactly the same as in the real setup.

IV. RESULTS AND DISCUSSION

The high-resolution profiles of the sensor's responses for the target force of 0.35 N are visualized in Figs. 5 and 6.

We note that during the production of the sensor array, the silicone encapsulating the barometric cells exhibited capillary action, resulting in a slightly concave surface. This concavity causes a decrease in contact area towards the outer cells of the real sensor, which results in increased pressure. For a better comparison with the actual flat sensor surface in simulation, we thus only focus on the three innermost cells.

Upon closer inspection of the sensor array (see Fig. 1), it becomes apparent that the positions of the orifices of the individual cells slightly differ and thereby do the distances between them. This is a consequence of the manual and, thus, imprecise work needed to enhance the barometric chips. We presume these discrepancies between cells to be the cause of slight discrepancies we see in the real-world data, e.g., slightly skewed and shifted sensor responses observable in Fig. 5 in cells 0 and 2.

During experiments with the simulated sensor, we recorded some measurements that displayed artifacts in the form of ridges and dents along the measured pressure curve for some taxels. This only happened if the probe tip was hitting the diagonal or edge of the sensor surface. Further investigation showed that the contact surface mesh, which usually remains constant, changes in these regions (see Fig. 7). The contact surface is constructed from the intersection of the volume mesh representing our sensor and the surface mesh representing the measuring probe. In the problematic regions, the probe tip intersects multiple tetrahedra of the volume mesh. This yields a more complex contact surface mesh with variations of pressure applied in each triangle, which causes tiny inconsistencies in our sensor's measurements. At the time of writing, we are not sure yet if these inconsistencies are caused by erroneous computation of the sensor's pressure values or stem from a flaw in the contact surface computation.

One workaround we used to remove the artifacts occurring on the diagonal and produce the measuring curve depicted in Fig. 6 was to combine two sensors in the same location.

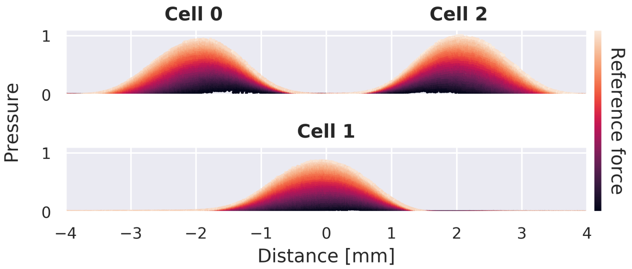

Fig. 5. Measured data from the real sensor. The sensor response for three adjacent cells on the path of the measuring tip relative to the sensor's center. The pressure was normalized using the maximum recorded pressure.

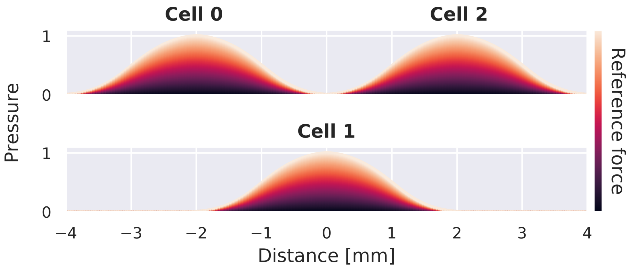

Fig. 6. Measured data from the simulated sensor. The sensor response for three adjacent cells on the path of the measuring tip relative to the sensor's center. The pressure was normalized using the maximum recorded pressure.

For one sensor, the volume mesh was rotated 90° around the z-axis. The pressure output of the combined sensor was then composed as the maximum pressure output of the two separate sensors.

V. Conclusion

We find that the simulated tactile sensor provides a nearly perfect approximation of the real-world sensor, producing very similar responses. We note that this framework is not limited to the presented barometric sensor. Its architecture offers an easy way to realize new windowing functions for distinct filtering and adding noise models for sensor cells more prone to hysteresis.

The code of our MuJoCo-ROS framework[1] and the contact surfaces plugin[2] are open source and available on GitHub.

VI. Future Work

The current state of this project leaves room for optimization and extension in 3 domains:

a) Computational Cost: Computing the contact surfaces and sensor values is rather expensive, and we will try to use parallelization for faster execution times.

b) Accuracy: Receptive fields of the cells might extend over their boundaries. Thus, applying weighted distance functions over cell boundaries, i.e., strided 2D convolution on the sensor surface, could be beneficial to accurately model this behavior.

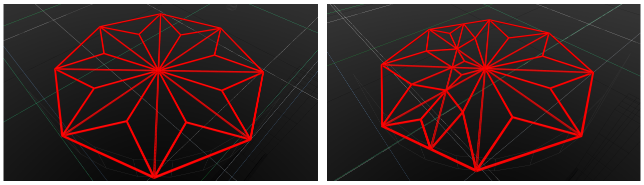

Fig. 7. Visualization of the contact surface mesh. The right image shows the contact surface when the measuring probe is in contact with the sensor diagonal. The left plot shows the contact surface at any other position on the sensor.

c) New Sensor Types: We are already working on an abstraction of taxels to non-flat surfaces to model curved sensors like the fingertip sensors deployed on our Shadow Dexterous Hand [10].

References

[1] N. Fazeli, S. Zapolsky, E. Drumwright, and A. Rodriguez, "Fundamental limitations in performance and interpretability of common planar rigid-body contact models," in *Robotics Research* (N. M. Amato, G. Hager, S. Thomas, and M. Torres-Torriti, eds.), (Cham), pp. 555–571, Springer International Publishing, 2020.

[2] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, IEEE, 2012.

[3] R. Elandt, E. Drumwright, M. Sherman, and A. Ruina, "A pressure field model for fast, robust approximation of net contact force and moment between nominally rigid objects," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8238–8245, IEEE, 2019.

[4] R. Tedrake and the Drake Development Team, "Drake: Model-based design and verification for robotics," 2019.

[5] R. Kõiva, T. Schwank, R. Haschke, and H. Ritter, "Towards high-density barometer-based tactile sensor arrays," in *Workshop on New Advances in Tactile Sensation, Perception, and Learning in Robotics: Emerging Materials and Technologies for Manipulation (RoboTac 2019)*, 2019.

[6] J. Masterjohn, D. Guoy, J. Shepherd, and A. Castro, "Velocity level approximation of pressure field contact patches," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11593–11600, 2022.

[7] R. Kõiva, T. Schwank, G. Walck, M. Meier, R. Haschke, and H. Ritter, "Barometer-based tactile skin for anthropomorphic robot hand," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9821–9826, IEEE, 2020.

[8] T. Möller and B. Trumbore, "Fast, minimum storage ray/triangle intersection," in *SIGGRAPH 2005 Courses*, ACM, 2005.

[9] I. Wald, "On fast construction of sah-based bounding volume hierarchies," in *2007 IEEE Symposium on Interactive Ray Tracing*, pp. 33–40, IEEE, 2007.

[10] R. Kõiva, M. Zenker, C. Schürmann, R. Haschke, and H. J. Ritter, "A highly sensitive 3d-shaped tactile sensor," *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1084–1089, 2013.

[1] https://github.com/ubi-agni/mujoco_ros_pkgs

[2] https://github.com/ubi-agni/mujoco_contact_surfaces