

Neural Cellular Automata for Decentralized Sensing using a Soft Inductive Sensor Array for Distributed Manipulator Systems

Bailey Dacre, Nicolas Bessone, Matteo Lo Preti, Diana Cafiso, Rodrigo Moreno,
Andrés Fañña, and Lucia Beccai

Abstract—In Distributed Manipulator Systems (DMS), decentralization is a highly desirable property as it promotes robustness and facilitates scalability by distributing computational burden and eliminating singular points of failure. However, current DMS typically utilize a centralized approach to sensing, such as single-camera computer vision systems. This centralization poses a risk to system reliability and offers a significant limiting factor to system size. In this work, we introduce a decentralized approach for sensing and in a Distributed Manipulator Systems using Neural Cellular Automata (NCA). Demonstrating a decentralized sensing in a hardware implementation, we present a novel inductive sensor board designed for distributed sensing and evaluate its ability to estimate global object properties, such as the geometric center, through local interactions and computations. Experiments demonstrate that NCA-based sensing networks accurately estimate object position at 0.24 times the inter sensor distance. They maintain resilience under sensor faults and noise, and scale seamlessly across varying network sizes. These findings underscore the potential of local, decentralized computations to enable scalable, fault-tolerant, and noise-resilient object property estimation in DMS.

Index Terms—Multi-Agent Systems · Distributed Manipulator Systems · Decentralized Sensing · Soft Inductive Sensor · Neural Cellular Automata

I. INTRODUCTION

A Distributed Manipulation System (DMS) consists of numerous actuators, often arranged in a lattice topology, that work collaboratively to manipulate objects on its surface. Through coordinated use of many actuators, these systems can achieve precise positioning, orientation, and manipulation of objects. This collaborative approach offers capabilities that surpass those of systems relying on independent actuators, which are often constrained by limitations in, e.g., force generation, stabilization, and actuation range [1]. Due to their spatial distribution, DMS cover a large workspace for manipulation and the possibility of parallel manipulation [2], [3].

M. Lo Preti, D. Cafiso, and L. Beccai are with the Istituto Italiano di Tecnologia, Genova, 16163, IT (e-mail: matteo.lopreti@iit.it, diana.cafiso@iit.it, lucia.beccai@iit.it).

B. Dacre, N. Bessone, R. Moreno, and A. Fañña are with the IT University of Copenhagen (ITU), København, DK (e-mail: baid@itu.dk, nbes@itu.dk, rodr@itu.dk, anfv@itu.dk)

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

Accurate knowledge of an object’s location is crucial for orchestrating manipulation in a DMS using closed-loop control [4]. Estimation of an object’s Geometric Center (GC) provides a low-dimensional representation of its position, equivalent to the center of mass for homogeneous objects. This information is critical for accurate manipulation through the application of external force. Many DMS utilize a centralized sensing system, such as single-camera computer vision systems [5]. However, centralized architectures introduce vulnerabilities: failure of a single sensor can compromise the entire system. Moreover, such sensing systems are external to the manipulator, and the need for specialized processing hardware for signal processing (e.g. FPGA), impairs their integration greatly [6].

Decentralization offers a promising alternative for DMS, addressing many of the limitations of centralized systems. By distributing the computational load of sensing and control across multiple components, decentralization mitigates the scalability challenges posed by centralized control, which often suffer from computational, processing, and communication bottlenecks [7]. Additionally, decentralization enhances robustness by eliminating the existence of a single point of failure. However, decentralized systems face their own challenges, particularly in coordinating interactions between agents. Inter-agent communication overhead can lead to latency and complicate real-time interactions among manipulators.

Localized sensing can provide precise information about object-manipulator interactions at specific known points. Information of such interactions is key for accurate manipulation through controlled force application. However, such locality is absent in camera based perception system. When localized sensors are utilized in a high spatial density, the system is able to detect objects in contact with multiple sites simultaneously, allowing for calculation of local distributions of the sensed modality, for example pressure maps. The use of multiple sensors also enhances system robustness by preserving functionality, even in the case of individual sensor failures or noise. Despite this, existing DMS typically implement one sensor per actuator in low density, discontinuous arrangements, limiting their ability to capture global object dynamics effectively.

Recognizing the critical need for decentralization, scalability, and robustness in DMS, this work offers two main contributions. First, We propose a novel inductive sensor design tailored for distributed sensing, offering a scalable, fault-tolerant approach

to sensing. Second, we investigate how an Neural Cellular Automata (NCA) based system utilizing this sensor can estimate global properties from purely local information, and how such a system is scale invariant and robust to fault, both key factors for producing DMS of any size.

II. RELATED WORKS

Advancing the capabilities of DMS requires addressing challenges in sensing and control architectures. This section reviews existing approaches to sensing in DMS and explores the potential of NCA as a scalable and robust solution for decentralized systems.

A. Sensing in Distributed Manipulation Systems

DMS have traditionally relied on open-loop control for object manipulation [8], [9], [10], [11]. Such systems often approximate programmable vector fields to achieve sensor-less object manipulation [12], [13], [14].

When sensing is incorporated, single-camera systems remain the most prevalent approach. Cameras provide rich environmental information, enabling object localization, pose estimation, and obstacle avoidance. Their widespread availability, large sensing area, and relatively high information density make them an attractive choice for DMS [2], [3], [9]. Beyond standard RGB cameras, some DMS incorporate depth cameras [15], depth cameras and RGB cameras [16], or marker-based systems [17] for accurate 3D tracking, particularly pertinent for systems capable of large vertical displacement.

In contrast, other DMS utilize localized sensing mechanisms to detect object interactions. Resistive tactile force sensors mounted on the manipulator end-effector are commonly used to measure contact forces, offering a robust and cost-effective solution [18], [19]. Photodiodes [20], [6], [21] and proximity sensors [22] have also been employed to detect position of objects. Some systems combine local sensing with external camera-based perception to fuse modalities for enhanced object tracking [23]. The low cost of tactile sensors makes them viable for high-density deployment, facilitating spatially distributed sensing. However, current DMS designs often employ low-density, discontinuous sensor configurations, typically using a single sensor per actuator, which restricts their capacity to effectively capture the object dynamics in detail.

Effective DMS sensing systems should provide precise local sensing with high spatial density to accurately capture global object dynamics. Scalability demands cost-effective sensors for large-scale deployment, along with architectures capable of handling inevitable sensor failures. In this work, we propose such a system utilizing a novel inductive smart sensor surface, able to provide high precision localized sensing. We then implement a Neural Cellular Automata (NCA) based architecture, facilitating a purely decentralized sensing framework, that is robust to failures and capable of scaling to any scale.

B. Neural Cellular Automaton

Cellular Automata (CA) consist of a regular grid of cells, each occupying one of a limited set of states. The state of a

cell is updated based on its own current state and that of its neighbors, according to predefined rules. Despite the simplicity of the local rules, CA have been shown to give rise to very complex emergent behaviors, exemplified by Conway's Game of Life [24]. However, crafting rules for specific behaviors is non-trivial, prompting a shift toward discovering rule sets through automated approaches [25].

Recently, deep learning has been integrated into CA to learn rule sets that drive desired behaviors through gradient-based optimization of a loss function [26], [27]. Neural Cellular Automata merge CA principles with Neural Network (NN), representing the CA update function f_θ as a network taking as input the agents neighborhood. A unique feature of NCAs is the use of hidden channels in each cell's state in which free tokens of information are stored for inter-agent communication [28], [29].

NCAs have been applied to a variety of tasks requiring global property estimation from local interaction. For instance, [30] demonstrated the use of NCAs to classify MNIST digits through consensus among agents. Similarly, [31] proposes a shape-aware controller where each module infers the shape of a larger assembly. [32] extended this concept to modular robotics, enabling systems to infer their own shapes through local communication, and successfully transitioned from simulation to hardware. [33] introduced a methodology for inferring the geometric center of objects laying on a grid of sensing agents. Leveraging the capabilities of NCAs, each agent locally shares information within its neighborhood, enabling the inference of global properties through purely local communication. This paper builds upon these findings to address the challenges of decentralized sensing in DMS, introducing a system that bridges the gap between simulation and real-world hardware.

III. SOFT INDUCTIVE SENSING PLATFORM

To satisfy a DMSs requirements for high precision local sensing at the actuator level, we developed a soft inductive sensing layer that serves dual purposes: enabling object manipulation as the system's end effector and collecting detailed tactile data about objects interacting with the surface.

A. Design Overview

The sensor design employs inductive sensors positioned bellow a compliant surface embedded with ferromagnetic material. Deformation of the soft material under load changes the distance between sensor and ferromagnetic material, causing a measurable change in inductance proportional to deformation. A Printed Circuit Board (PCB), containing a lattice of coils connected to inductive signal conditioning chips, form an array of inductive sensors. This PCB forms the base layer for a soft structure comprising a ferromagnetic sheet encapsulated within lightweight, compliant polyurethane foam (Poron®) and topped with a smooth, FDA-approved Polyethylene Terephthalate (PET) film. This single soft structure allows these sensors to be combined into a continuous soft sensing surface. This surface is well suited as a sensorized DMS end-effector, and for industrial applications like food handling and packaging,

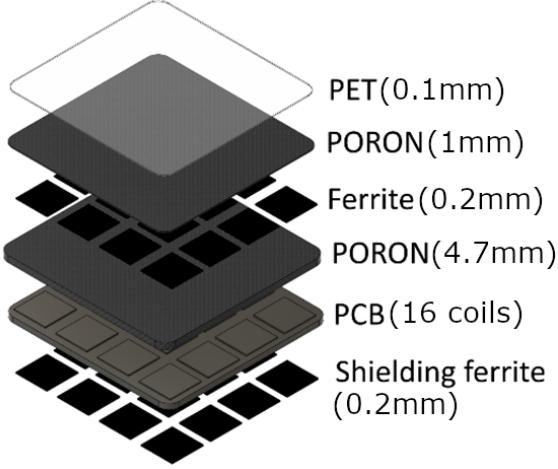


Figure 1: Exploded schematic of the soft sensor prototype, showing the PCB, ferrite, Poron®, and PET layers.

due to its ruggedness and resilience against dust and moisture. This architecture allows for low-cost scalability. The number, arrangement, shape, and spacing of coils within each PCB configurable to application need.

B. Prototype board design and manufacture

The sensor design utilizes a LDC1614 inductive signal conditioning chip and paired coil as an inductive sensor, following the procedure in [34]. For use in our experimental testing, a FR4 PCB hosting 16 embedded inductive coils arranged, each measuring 30×30 mm, were arranged in a 4×4 grid. The tactile sensor prototype uses four LDC1614 connected to two I2C lines, with each chip polling four coils and two chips per I2C line. This configuration allows precise measurements at up to the kHz range. To optimize performance, traces between the coils and driver chips are isolated to enhance the signal-to-noise ratio, and the board is shielded from electromagnetic interference using an additional ferrite layer beneath the PCB. In this prototype, we connect the sensors to a Teensy® micro-controller from which to read sensor data via I2C.

The sensor's structure and material selection is illustrated in fig. 1. The multilayer tactile structure above the PCB integrates two Poron® foam layers (1mm and 4.7mm respectively) with a ferrite sheet (0.2mm) to maximize compliance and sensitivity. The layers are bonded with Sil-Poxy® adhesive. The PET top layer (0.1mm) provides a smooth, low-friction contact surface, ideal for handling soft materials and reducing wear. Laser-cut plexiglass masks were used for precise assembly and to ensure manufacturing consistency.

C. Characterization and Testing

A three-axis indentation setup tested the tactile sensing layer prototype. Two micrometric manual linear stages controlled the positions of the X and Y stages. An M-111.1DG translation stage was positioned along the Z-axis on top of the X stage, controlled by the paired C-884.4DC motion controller

(Physik Instrumente, USA). An ATI Nano17 (ATI Industrial Automation, USA) load cell was mounted on the Z stage with an L-shaped part to adjust its configuration, under which an ABS probe with a round tip shape was attached to the load cell. The prototype was placed on a lab jack (MKS Instruments, Inc.) and fixed with a base designed to locate the four testing positions.

Key performance metrics, including repeatability, range, crosstalk, RMSE, hysteresis, and sensitivity, were analyzed for all 16 inductive coils under controlled loading conditions. Repeatability in all coils was measured at 68.22 % with a standard deviation of 27.33 %. The maximum force range was $6.72 \text{ N} \pm 0.78 \text{ N}$, and crosstalk between neighboring coils was minimal, measured at $1.57 \% \pm 1.37 \%$.

To generate calibration curves, we applied a uniform pressure to the (30 mm^2) area above each sensor via incremental loading, applying between 0.04905 - 5.886 N of force, equivalent to 5-600 g of applied mass. Polynomial curve fitting per coil found each coil's response to be predominately linear, though distinct due to edge effects effects and inbuilt manufacturing variability.

IV. DECENTRALIZED SENSING

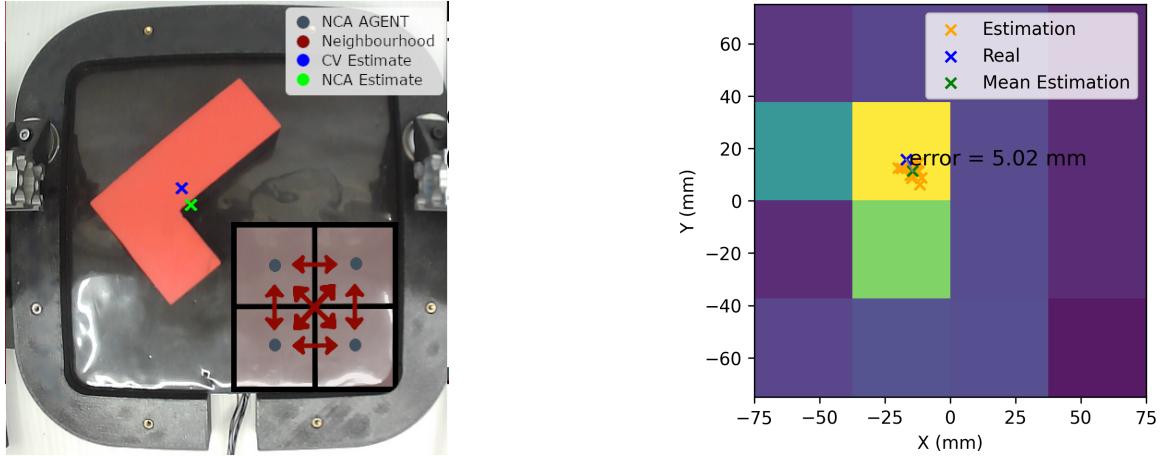
In this work, we estimate the global properties of objects in contact with the tactile sensing layer, specifically the GC of the surface of an object in contact with the sensor. For objects of uniform density, this corresponds to the 2D projection of their Center of Mass (CoM).

A. Data collection

To train the NCA model, a dataset was created containing two key components: readings from the sensors when objects were in contact with the sensing surface, serving as input, and the ground truth geometric center of each object, serving as the target output. The ground truth was determined using a computer vision system developed for this purpose using the OpenCV library [35].

The dataset consisted of distinct geometric objects with uniform mass distribution but varying shape and mass, as detailed in fig. 3. During data collection, a predetermined face of each object was placed in direct contact with the sensor surface. Sensors readings were sampled at 20 Hz for 2.5 seconds to produce 50 samples per object per position. Between 50-150 positions were recorded for each object, depending on object size relative to sensor area, across the entire sensor surface.

To ensure reliable detection by the computer vision system, all objects were 3D printed from bright, mono-colored PLA. Edges of the objects not in contact with the sensor or relevant for detection were masked with black tape. Images were captured under controlled lighting conditions, and the OpenCV Python library was employed to calculate the geometric center of the face of the object in contact with the sensor. The geometric center was then mapped to the coordinate frame of the sensor board, providing ground truth data for training.



(a) Geometric center of object detected by computer vision system (blue) and calibrated NCA estimate, projected to object top surface (green). Visualization of neighborhood (red) of agents (gray).

(b) Estimates for each of the NCA agents (orange), their mean (green), and object center detected by computer vision system (blue)

Figure 2: Estimation of Geometric Center for object in contact with sensing surface

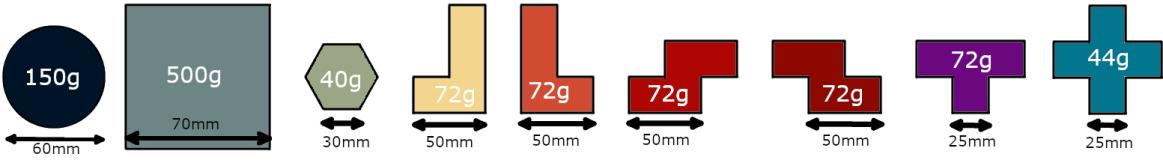


Figure 3: Contact footprint and mass of objects used to create dataset.

B. Neural Network Model

A decentralized system was implemented, where an array of NCA agents received inputs directly from individual sensors. The spatial distribution of the sensors, such as those within the sensor board mirrors the lattice structure of a 2D CA, enabling decentralized and spatially distributed sensing.

Although the sensors were implemented within the same sensor board for manufacturing simplicity, each NCA agent could only access the local information of its corresponding sensor and its neighbors. This collectively forms a distributed network where the agents rely solely on local information; a fully decentralized computational paradigm.

In the NCAs framework, each agent maintains a state S that evolves iteratively through an asynchronous update process governed by a NN-based update function. The state S of a tile i at time t is updated according to:

$$S_i^{t+1} = f_\theta(\{S_j^t\}_{j \in N(i)}) \quad (1)$$

where f_θ is the additive update function parameterized by θ , this function takes as input the states of tile i and its neighborhood $N(i)$ from the previous time step $t - 1$, enabling localized updates influenced by both the tile itself and its neighbors.

1) *Agents State:* The state S of each NCA agent encapsulates multiple components: the sensor value V , which captures tactile interactions with the environment; the global property

estimation E , representing the agent's prediction of the global property; a set of hidden channels H , which serve as auxiliary memory or communication channels; and information from its neighborhood N , which encodes the states of the tiles within the agent's Moore's neighborhood, as illustrated in Fig. 2. The neighborhood N is restricted to immediate neighbors and does not extend to the neighbors of neighbors.

During training, the NN modifies E to minimize the prediction error, while the dynamics of H are left to emerge as the network optimizes its functionality. In this framework, global consensus emerges iteratively through local exchanges, with agents requiring multiple update steps to converge. The number of iterations is randomly sampled from a uniform distribution in the arbitrary range of 15-30 time steps to ensure robustness to long-term stability issues, as described in [36].

In a distributed setting, a shared global clock cannot be assumed, in such scenarios, the agents update their states asynchronously. During each training step, only a randomly selected subset of agents updates its state. Once the predetermined number of iterations is reached, the estimation error is calculated as the mean Euclidean distance between the predicted center of the object (x_{Ei}, y_{Ei}) for each agent and the actual center (x_C, y_C) determined via a computer vision model. Implementation and reproducibility kit available in the footnote ¹.

¹<https://github.com/nhbess/NCA-REAL>

2) *Architecture*: The update function f_θ is implemented as a NN with three main layers: The Perception Layer, which applies a 3×3 convolutional kernel to extract local features, and a Sobel filter to compute gradients of the states along the x and y axes; a Processing Layer utilizing a 1×1 kernel to reduce dimensionality and extract relevant features, with a with a Rectified Linear Unit (ReLU) to introduce linearity; finally, the Output Layer, also employing a 1×1 kernel, generates residual updates to the agent's state, modifying only the global property estimation and hidden channels while preserving other components of the state.

3) *Training Methodology*: Training the NCA involves learning the parameters θ of the update function f_θ to ensure that the estimated global property E converges to its true value. All agents in the system are identical and share the same neural network. The was randomly divided into two equally sized distinct sets: training and testing. The key variable of interest, the estimation E , is used to compute the loss function by comparing agent estimation to the true center of the object derived from the dataset.

To enhance stability, the training incorporates a pool-based strategy, in which poorly performing states are periodically replaced with empty states from the pool, as detailed in [29]. This approach mitigates training instability and ensures robust performance. The efficacy of this methodology has been demonstrated previously [33], validating its application in distributed sensing.

V. EXPERIMENTS

To assess the performance of our system, in context of a DMS as previously described, we conducted a series of experiments. First, we aim to quantify the accuracy of the decentralized estimation of the global property: the geometric center. Additionally, we investigate the system's robustness, which we categorize into two distinct aspects: fault tolerance and noise tolerance. Fault tolerance pertains to the system's ability to maintain functionality when individual components fail or are removed, while noise tolerance evaluates its resilience to variations or noise in sensor signals. Lastly, we explored the scalability capabilities of the model by testing its performance across sensor arrays of varying sizes.

A. Performance Evaluation

In this experiment, we compared the performance of two models, trained using calibrated and uncalibrated data, respectively, as detailed in section IV-B. Both models were tested on their respective datasets, and the average estimation provided of individual agents was computed with the ground truth. Figure 4 shows that both models exhibit an average distance error of 7.19 mm and 6.79 mm respectively.

This result highlights that the system can effectively estimate the GC with a high degree of accuracy. Given the coil width of 30 mm and a center-to-center coil spacing of 37.5 mm, achieving a positional error of below 7.2 mm (≈ 0.24 times the width of a coil) showcases the system's ability to accurately determine the GC at much smaller distances than coil spacing

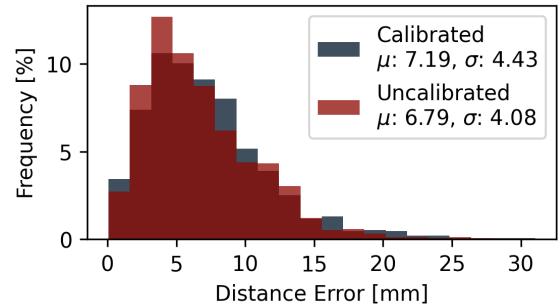


Figure 4: Error distribution of models trained and evaluated with calibrated and uncalibrated data from respective test sets.

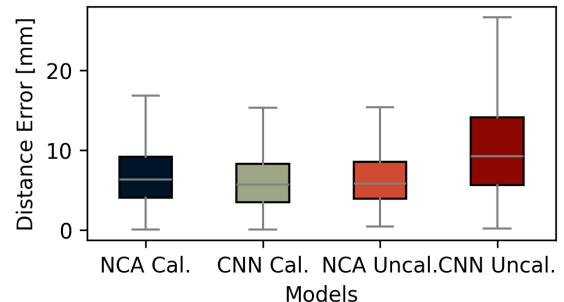


Figure 5: Box plots of estimation error for Centralized and NCA approaches, showing median (line), inter-quartile range (box), and range (whiskers)

of without relying on it being positioned directly above a single coil.

The use of uncalibrated data appeared to slightly reduce estimation errors compared to the use of calibrated data. However, after performing statistical testing, the distributions were found not to be statistically significant, with $P = 0.304$.

B. Comparison with Centralized Approaches

To evaluate the performance of the proposed NCA-based system, we compare it against a centralized Convolutional Neural Network (CNN) architecture. The centralized NN processes all sensor readings simultaneously using two consecutive convolutional layers to extract global spatial features, followed by 3 fully connected layers to estimate the object's position. The resulting comparison is shown in fig. 5, where it can be observed that both the centralized and decentralized approaches perform comparably on calibrated data. Notably, the uncalibrated centralized model exhibits slightly worse performance, potentially due to domain shifts that the architecture cannot accommodate without calibration.

C. Fault Tolerance

Fault tolerance was evaluated by systematically introducing sensor failures and measuring the system's performance. Faulty sensors were simulated by replacing their readings with zero, effectively removing their influence on the system's computations. We masked a fault on between 0% to 90%

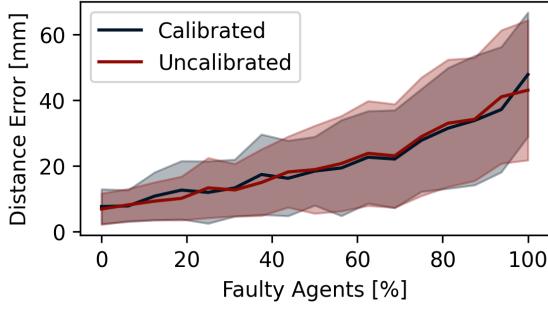


Figure 6: Mean and standard deviation of estimation error for different percentages of faulty agents.

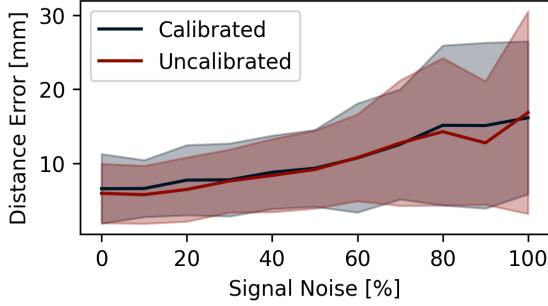


Figure 7: Mean and standard deviation of estimation error for different noise levels.

of sensors (rounded down) in increments of 10%. The model was then evaluated across 100 different object locations, each time with a new random selection of faulty sensors to ensure statistical robustness. We measured the distance error between the estimated GC and the ground truth, recording both the mean and standard deviation of these errors, the results are shown in [fig. 6](#).

D. Noise Tolerance

The system’s noise tolerance was evaluated by introducing Gaussian noise to sensor data at varying levels, ranging from 0% to 100% in increments of 10%. The noise magnitude was scaled relative to the sensor’s signal strength, simulating real-world conditions such as electronic interference, signal attenuation, or sensor drift.

For each noise level, 100 trials were conducted, and the mean and standard deviation of the distance error were recorded. As shown in [fig. 7](#), the system demonstrates resilience to increasing noise levels, maintaining consistent accuracy at moderate noise intensities and gracefully degrading as noise becomes severe.

E. Scalability

One of the key advantages of NCAs is their inherent locality, allowing the same trained model to be applied to networks of varying sizes without retraining.

To evaluate this property, a model trained on a grid of 8×8 sensors was tested on grids ranging from 4×4 to 100×100 , representing up to 10000 agents. Due to hardware

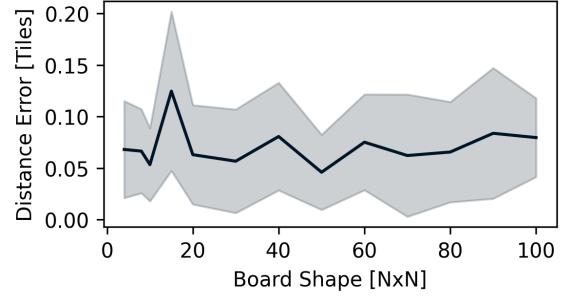


Figure 8: Mean and standard deviation of estimation error for different sizes of agents network.

limitations, binary synthetic data was used for both training and evaluation. Performance was measured in terms of distance errors expressed in tile sizes (the distance between adjacent sensors). [Figure 8](#) shows the results of the experiment.

The mean distance errors remained consistent across all network sizes. Error metrics did not increase with the addition of more agents, indicating that the model successfully scaled without any degradation in performance. This consistency highlights the effectiveness of NCAs in maintaining robustness and accuracy, regardless of the network’s size.

VI. DISCUSSION

A. Centralized vs Decentralized sensing

There are fundamental trade-offs between centralized and decentralized architectures for global property estimation. Our decentralized NCA-based approach performs comparatively to a centralized baseline approach. Centralized systems necessitate all sensor data is aggregated, creating a computational bottleneck and a single point of failure, which our system avoids. However, this does allow rapid access to information from across the system, without the need to wait for information propitiation via local communication, which is costly both in time and computational overhead. This delay can be seen in the non-instantaneous nature of generating our system’s estimate. Therefore, our system may struggle to achieve optimal results in the case of highly dynamic environments where object properties of position varied rapidly.

Decentralized perception systems also face complexity challenges, as they lack direct access to global information and cannot implement straightforward global algorithms for property estimation. While decentralized variants of such algorithms have been explored [21], they incur additional computational costs. Despite these challenges, the scalability and robustness inherent to decentralized architectures make our NCA-based approach a promising candidate for large-scale DMS, particularly in harsh environments with high noise levels or a high likelihood of sensor failure.

B. Sensor Calibration

The similarity of models trained on both calibrated and uncalibrated sensor data suggests that, in this context, the calibration process may not substantially influence the system’s

estimation capabilities. If calibration is unnecessary, this reduces the complexity of setup for any practical deployments. However, although we have shown this state to be the case in this context, this state does not necessarily hold in general. Sensors manufactured with a higher manufacturing variability (i.e. not manufactured as part of the same sensor board) or operating in substantially different conditions, which are not accounted for without calibration. Soft sensors may be especially susceptible to these variations, due to the large mechanical deformations they experience in use.

Results from the noise tolerance experiments provide insights into valid operational conditions. By examining how the system responds to controlled variations in sensor input quality, it is possible to approximate the influence of different calibration distributions on estimation error, thus providing a measure of confidence in the model's robustness when used with diverse sensor configurations.

C. System Robustness

The system's fault tolerance experiment reveals that meaningful performance is maintained even when up to 30% of the sensors are rendered non-functional, and a robustness up to a 50% signal-to-noise ratio indicates a reliability to the system. In any real world implementation, such noise and sensor failures are inevitable, especially as the number of sensors increases. Such deviation in sensor readings would also be expected on a non-static system, as in a DMS actuator, in which measured pressure applied to the sensor board would vary as the system moves. Therefore, this consistency in estimation offers validation for the utility of such a system for real world application. The removal of a centralized processing unit significantly enhances robustness by eliminating a single point of failure. The consensus mechanism inherent to NCAs ensures that the system remains robust even when individual agents receive corrupted information. This resilience to sensor failures and noisy inputs is critical for real-world applications, where unpredictable environmental factors and hardware degradations are common.

D. Scalability

One of the most notable advantages of the proposed NCA-based approach is its scalability. The system's performance remains consistent across varying network sizes by relying solely on local information and decentralized decision-making. Unlike centralized architectures, this system imposes no size limitations due to computational overhead, allowing for theoretically unlimited scaling. However, as the number of agents increases, so does the time for information to propagate via local communication. This limitation could potentially be exploited, prioritizing sensing and manipulation for objects in the immediate vicinity of an agent in the case of multi-object manipulation. Additionally, the ability to implement a single agent on multiple system scales without need for retraining offers a massive benefit for practical deployment. This locality also indicates such a architecture would be well

suited for modular designs, where the sensing surface can be easily adapted with little need for model retraining.

VII. FUTURE WORK

A. Non-static objects and surfaces

While the experiments in this study focused on static objects, real-world applications in object manipulation involve dynamic environments. A static approximation may hold with a sufficiently high sampling rate relative to object speed. However, challenges arise from the time required for information propagation and consensus formation in decentralized systems. When integrated as a robotic end effector, the sensor will be non-static and will experience induced force, altering estimation. Even if held static, the surface will experience pose variations that influence contact forces. While traditional mechanics can estimate these forces, additional complexities, such as soft-surface compression under non-perpendicular orientations to gravity, must be considered. Although noise robustness testing offers some insight into the behavior behavior of the current system, addressing these effects will require further experimentation and enhanced model training.

B. Tunable Material

The materials of the soft surface influence both sensor characteristics and manipulation capabilities. Tailoring these properties for specific applications, such as optimizing deformation to maximize the dynamic sensing range for objects with known properties, holds significant potential. Furthermore, varying material properties across the surface could enable new sensing and manipulation strategies, which we intend to explore in future work.

VIII. CONCLUSION

This work presents a decentralized strategy for sensing in Distributed Manipulation System (DMS) using Neural Cellular Automata (NCA) in combination with a newly developed soft inductive sensor. We demonstrated that this approach reliably infers global object properties, such as the geometric center, using purely local interactions. The system exhibits robustness to noise and sensor faults, maintaining performance under challenging conditions. Moreover, we have shown that these models scale without loss of accuracy to significantly larger sensor arrays, reinforcing the potential of decentralized methodologies to overcome the limitations of centralized sensing and control. This proof of concept lays the foundation for fully distributed, fault-tolerant manipulation architectures capable of dynamically responding to component failures. The findings suggest a promising direction for future research in developing versatile, scalable, and efficient DMS, particularly for applications in harsh environments or those requiring high reliability and adaptability.

A. Acknowledgments

This work was conducted as part of the MOZART project, funded by the European Union, EU project id: 101069536.

REFERENCES

- [1] K. F. Böhringer and H. Choset, Eds., *Distributed Manipulation*. Boston, MA: Springer US, 2000. [Online]. Available: <https://link.springer.com/10.1007/978-1-4615-4545-3>
- [2] D. Reznik and J. Canny, "C'mon part, do the local motion!" in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, vol. 3. Seoul, South Korea: IEEE, 2001, pp. 2235–2242. [Online]. Available: <http://ieeexplore.ieee.org/document/932955/>
- [3] M. Ataka, B. Legrand, L. Buchaillot, D. Collard, and H. Fujita, "Design, Fabrication, and Operation of Two-Dimensional Conveyance System With Ciliary Actuator Arrays," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 1, pp. 119–125, Feb. 2009. [Online]. Available: <http://ieeexplore.ieee.org/document/4783214/>
- [4] J. E. Luntz, W. Messner, and H. Choset, "Distributed Manipulation Using Discrete Actuator Arrays," *The International Journal of Robotics Research*, vol. 20, no. 7, pp. 553–583, Jul. 2001.
- [5] T. D. Murphey and J. W. Burdick, "Feedback Control Methods for Distributed Manipulation Systems that Involve Mechanical Contacts," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 763–781, Aug. 2004, publisher: SAGE Publications Ltd STM. [Online]. Available: <https://doi.org/10.1177/0278364904045480>
- [6] M. Ataka, M. Mita, and H. Fujita, "The layer-built sensor/actuator integrated array for the 2D feedback conveyance," in *2007 IEEE 20th International Conference on Micro Electro Mechanical Systems (MEMS)*, Jan. 2007, pp. 35–38, iSSN: 1084-6999. [Online]. Available: <https://ieeexplore.ieee.org/document/4433052>
- [7] P. K. Agarwal, L. E. Kavraki, and M. T. Mason, "Velocity Field Design for the Modular Distributed Manipulator System (MDMS)," in *Robotics: The Algorithmic Perspective*, 0th ed. A K Peters/CRC Press, Dec. 1998, pp. 45–58.
- [8] K.-F. Bohringer, V. Bhatt, and K. Goldberg, "Sensorless manipulation using transverse vibrations of a plate," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, vol. 2, May 1995, pp. 1989–1996 vol.2, iSSN: 1050-4729. [Online]. Available: <https://ieeexplore.ieee.org/document/525555>
- [9] I. Georgilas, A. Adamatzky, and C. Melhuish, "Cellular Automaton Manipulator Array," in *Robots and Lattice Automata*, ser. Emergence, Complexity and Computation, G. C. Sirakoulis and A. Adamatzky, Eds. Cham: Springer International Publishing, 2015, pp. 295–309. [Online]. Available: https://doi.org/10.1007/978-3-319-10924-4_13
- [10] K. Liu, F. Hacker, and C. Daraio, "Robotic surfaces with reversible, spatiotemporal control for shape morphing and object manipulation," *Science Robotics*, vol. 6, no. 53, p. eabf5116, Apr. 2021. [Online]. Available: <https://www.science.org/doi/10.1126/scirobotics.abf5116>
- [11] C. Liu, T. Tsao, Y.-C. Tai, W. Liu, P. Will, and C.-M. Ho, "A Micromachined Permalloy Magnetic Actuator Array for Micro Robotics Assembly Systems," in *Proceedings of the International Solid-State Sensors and Actuators Conference - TRANSDUCERS '95*, vol. 1, Jun. 1995, pp. 328–331. [Online]. Available: <https://ieeexplore.ieee.org/document/717185/?arnumber=717185&tag=1>
- [12] K.-F. Bohringer, B. Donald, R. Mihailovich, and N. MacDonald, "A theory of manipulation and control for microfabricated actuator arrays," in *Proceedings IEEE Micro Electro Mechanical Systems An Investigation of Micro Structures, Sensors, Actuators, Machines and Robotic Systems*. Oiso, Japan: IEEE, 1994, pp. 102–107. [Online]. Available: <http://ieeexplore.ieee.org/document/555606/>
- [13] K. Bohringer, B. Donald, L. Kavraki, and F. Lamiriaux, "Part orientation with one or two stable equilibria using programmable force fields," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 2, pp. 157–170, Apr. 2000. [Online]. Available: <http://ieeexplore.ieee.org/document/843172/>
- [14] S. Patil, T. Tao, T. Hellebrekers, O. Kroemer, and F. Z. Temel, "Linear Delta Arrays for Compliant Dexterous Distributed Manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 10 324–10 330, arXiv:2206.04596 [cs, eess]. [Online]. Available: <http://arxiv.org/abs/2206.04596>
- [15] S. Follmer, D. Leithinger, A. Olwal, A. Hogge, and H. Ishii, "inFORM: dynamic physical affordances and constraints through shape and object actuation," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*. St. Andrews Scotland, United Kingdom: ACM, Oct. 2013, pp. 417–426. [Online]. Available: <https://dl.acm.org/doi/10.1145/2501988.2502032>
- [16] C. Uriarte, A. Asphandiar, H. Thamer, A. Bengollo, and M. Freitag, "Control strategies for small-scaled conveyor modules enabling highly flexible material flow systems," *Procedia CIRP*, vol. 79, pp. 433–438, Jan. 2019, journal Abbreviation: Procedia CIRP Pages: 438 Publication Title: Procedia CIRP.
- [17] Z. Xu, C. Trakarnchaiyo, C. Stewart, and M. B. Khamesee, "Modular Maglev: Design and implementation of a modular magnetic levitation system to levitate a 2D Halbach array," *Mechatronics*, vol. 99, p. 103148, May 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957415824000138>
- [18] M. A. Robertson, M. Murakami, W. Felt, and J. Paik, "A Compact Modular Soft Surface With Reconfigurable Shape and Stiffness," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 16–24, Feb. 2019, conference Name: IEEE/ASME Transactions on Mechatronics. [Online]. Available: <https://ieeexplore.ieee.org/document/8514032/?arnumber=8514032>
- [19] Z. Xue, H. Zhang, J. Cheng, Z. He, Y. Ju, C. Lin, G. Zhang, and H. Xu, "ArrayBot: Reinforcement Learning for Generalizable Distributed Manipulation through Touch," Jun. 2023, arXiv:2306.16857 [cs]. [Online]. Available: <http://arxiv.org/abs/2306.16857>
- [20] A. Berlin, D. Biegelsen, P. Cheung, M. Fromherz, D. Goldberg, W. Jackson, B. Preas, J. Reich, and L.-E. Swartz, "Motion control of planar objects using large-area arrays of MEMS-like distributed manipulators," *Micromechatronics*, 2000.
- [21] M. D. Bedillion, D. Parajuli, and R. C. Hoover, "Distributed Sensing in Actuator Array Manipulation," in *Volume 4A: Dynamics, Vibration and Control*. San Diego, California, USA: American Society of Mechanical Engineers, Nov. 2013, p. V04AT04A002.
- [22] T. Fukuda, I. Takagawa, K. Sekiyama, and Y. Hasegawa, "Hybrid Approach of Centralized Control and Distributed Control For Flexible Transfer System," in *Distributed Manipulation*, K. F. Böhringer and H. Choset, Eds. Boston, MA: Springer US, 2000, pp. 65–85. [Online]. Available: https://doi.org/10.1007/978-1-4615-4545-3_4
- [23] D. Parajuli, M. D. Bedillion, and R. C. Hoover, "Actuator Array Manipulation Using Low Resolution Local Sensing," American Society of Mechanical Engineers Digital Collection, 2014, p. 8. [Online]. Available: <https://dx.doi.org/10.1115/IMECE2014-37147>
- [24] A. Adamatzky, *Game of Life Cellular Automata*. Springer, 2010.
- [25] S. Wolfram, "The problem of distributed consensus: a survey," *arXiv preprint arXiv:2106.13591*, 2021.
- [26] D. Ha and Y. Tang, "Collective Intelligence for Deep Learning: A Survey of Recent Developments," Mar. 2022.
- [27] W. Gilpin, "Cellular automata as convolutional neural networks," *Physical Review E*, vol. 100, no. 3, Sep. 2019.
- [28] N. Wulff and J. A. Hertz, "Learning Cellular Automaton Dynamics with Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 5. Morgan-Kaufmann, 1992.
- [29] A. Mordvintsev, E. Randazzo, E. Niklasson, and M. Levin, "Growing Neural Cellular Automata," *Distill*, vol. 5, no. 2, p. e23, Feb. 2020. [Online]. Available: <https://distill.pub/2020/growing-ca>
- [30] E. Randazzo, A. Mordvintsev, E. Niklasson, M. Levin, and S. Greydanus, "Self-classifying MNIST Digits," *Distill*, vol. 5, no. 8, p. e00027.002, Aug. 2020. [Online]. Available: <https://distill.pub/2020/selforg/mnist>
- [31] G. Nadizar, E. Medvet, K. Walker, and S. Risi, "A Fully-distributed Shape-aware Neural Controller for Modular Robots," in *Proceedings of the Genetic and Evolutionary Computation Conference*. Lisbon Portugal: ACM, Jul. 2023, pp. 184–192.
- [32] K. Walker, R. B. Palm, R. Moreno, A. Faina, K. Stoy, and S. Risi, "Physical Neural Cellular Automata for 2D Shape Classification," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Kyoto, Japan: IEEE, Oct. 2022, pp. 12 667–12 673. [Online]. Available: <https://ieeexplore.ieee.org/document/9981214/>
- [33] N. Bessone, P. Zahadat, and K. Stoy, "Neural Cellular Automaton for Decentralized Inference in Distributed Manipulation Systems," in *Advances in Practical Applications of Agents, Multi-Agent Systems, and Digital Twins: The PAAMS Collection*, P. Mathieu and F. De la Prieta, Eds. Cham: Springer Nature Switzerland, 2025, pp. 61–72.
- [34] M. Lo Preti and L. Beccai, "Sensorized objects used to quantitatively study distal grasping in the African elephant," *iScience*, vol. 26, no. 9, p. 107657, Sep. 2023. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S2589004223017340>
- [35] G. Bradski, *The OpenCV Library*, Nov. 2000, vol. 25, journal Abbreviation: Dr. Dobb's J. Softw. Tools Publication Title: Dr. Dobb's J. Softw. Tools.
- [36] S. Wolfram, "Universality and Complexity in Cellular Automata."