Linear Delta Arrays for Compliant Dexterous Distributed Manipulation

Sarvesh Patil, Tony Tao, Tess Hellebrekers, Oliver Kroemer, F. Zeynep Temel

https://iamlab-cmu.github.io/delta-arrays/

Abstract— This paper presents a new type of distributed dexterous manipulator: delta arrays. Our delta array setup consists of 64 linearly-actuated delta robots with 3D-printed compliant linkages. Through the design of the individual delta robots, the modular array structure, and distributed communication and control, we study a wide range of in-plane and out-of-plane manipulations, as well as prehensile manipulations among subsets of neighboring delta robots. We also demonstrate dexterous manipulation capabilities of the delta array using reinforcement learning while leveraging compliance. Our evaluations show that the resulting 192 DoF compliant robot is capable of performing various coordinated distributed manipulations of a variety of objects, including translation, alignment, prehensile squeezing, lifting, and grasping.

Index Terms— Multi-Robot Systems, Soft Robot Applications, Dexterous Manipulation

I. INTRODUCTION

The term *dexterous manipulation* often invokes the image of a five-fingered hand delicately holding an object as a human would. However, robots are not restricted to human morphology. Imagine instead a surface covered in fingers. Each finger can move its fingertip in a small 3D workspace above its fixed base and interact with parts of objects that enter its workspace. The fingers can work together to shift, tilt, lift, block, and even pinch objects. The large number of fingers provides additional redundancy, with larger objects being manipulated by tens of fingers at a time. The distributed nature of the fingers also means that multiple objects can be easily manipulated in parallel in different regions of the surface. This type of system would thus represent a distributed dexterous manipulation paradigm.

In this paper, we present an array of linear delta robots for the development of distributed dexterous manipulation strategies. Delta arrays consist of grids of small prismatic soft delta robots (3 degrees-of-freedom (DoF) each) that work together to manipulate objects. We propose a modular design for the delta arrays that consist of 2×2 units (12 DoF each) with each unit having a standalone mechanical and electronic design. Each unit has its own processor and controllers, allowing for distributed computation with a central computer providing high-level commands. We also present a real hardware implementation of an 8×8 array consisting of 16 units and providing 192 degrees of freedom.

S. Patil, T. Tao, O. Kroemer, and F. Z. Temel are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA {sarveshp, longtao, okroemer, ztemel} @andrew.cmu.edu

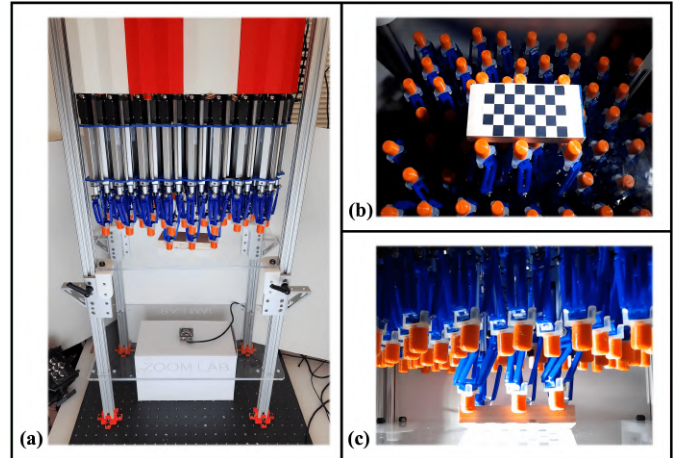Tess Hellebrekers is with Meta AI Research tessh@fb.com

Fig. 1: Delta array dexterous manipulation setup with robots facing down. (a) The setup consists of 64 delta robots actuated by linear motors and a camera. (b) Checkered board on the object used for pose estimation. (c) Distributed manipulation strategy for tilting wooden block.

Each compliant delta robot in the array is actuated by three linear actuators. These actuators are connected via parallel mechanisms to an end-effector platform. The platform and parallel linkages are 3D printed together out of thermoplastic polyurethane (TPU - 95A shore hardness) for easier assembly, compliant interactions, and low hysteresis under extreme deformations. The linear-actuator design allows for the delta robots to be packed closely together, in a hexagonal grid, and for their end-effectors to move outside of the footprint of the actuators. This allows the workspaces of neighboring deltas to overlap, and perform prehensile manipulations such as pinching between neighboring delta robots.

We present two modes of operating the delta array:

- Facing Down - Objects are placed on a plexiglass plane and manipulated from the top with a camera underneath the array for visual feedback (Fig. 1).
- Facing Up - Objects are placed on top of the array and manipulated from underneath or on the sides (Fig. 5).

The delta array provides a basis for a wide range of different manipulation strategies. Similar to smart conveyors, delta arrays are capable of executing various planar transportation behaviors. Unlike smart conveyors, delta arrays need to use a finger gaiting approach, with coordinated making and breaking of contacts across delta robots, to shift objects across the array's workspace. This added complexity, however, means delta arrays can make better contact with
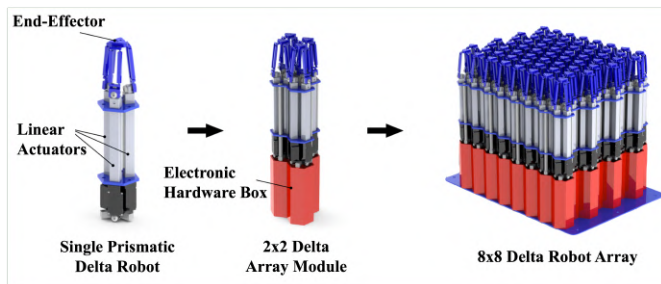
Fig. 2: The modular design of the delta array. Each robot consists of three linear actuators and a 3D printed TPU end effector (left). Four robots organized in a 2×2 hexagonal grid form a module, which shares the electrical components (middle). 16 modules form the delta array (right).

objects that have non-planar surfaces. The additional flexibility and non-planar motions of the deltas also allow for a number of other strategies that require out-of-plane motions in 3D workspaces. These strategies contrast with traditional manipulators in the sense that the end-effectors are static as opposed to being mounted on a robotic arm, and rely more on cooperation between multiple agents to accomplish tasks.

In the facing-up mode, the variable height allows for the rolling and tilting of objects on the array surface. Fingers can be raised to create fixture-like structures for aligning objects. The lateral motions allow the deltas to grasp and pinch objects of various sizes across the array. Although many of these strategies have been individually supported by other distributed or dexterous manipulation systems, to the best of our knowledge, this is the first system that supports all of these strategies and thus the possibility of combining strategies, as well as learning new ones.

Implementing and controlling an array of delta robots presents a number of challenges. The design needs to be modular for easy construction, extension, and maintenance. The individual delta robots need to be robust and safe, but also precise and capable of supporting a wide range of manipulation strategies. The communication needs to be fast and scalable to minimize command latency throughout the array network. In the remainder of this paper, we will explain and discuss our design decisions in developing the delta arrays and how we tackle each of these systemic requirements. Sections II, III, and IV describe the modular arrays in a bottom-up manner, while sections V, and VI present basic control strategies for manipulating objects with the arrays. The focus of this paper is on the design of the delta arrays and demonstrating its ability to execute a variety of distributed dexterous manipulation strategies. Developing more advanced and hybrid strategies is left to future work.

II. RELATED WORK

A. Delta robots

Delta robots were introduced by Clavel in 1990 and initially designed as a pick-and-place tool [1]. Conventional delta robots have a fixed base and a moving stage that are always parallel to each other. These platforms are connected by three kinematic chains with revolute and universal joints.

These chains are each driven by single-DoF actuators that are positioned at the fixed base. The motion is transmitted from the base arm to the moving stage by three parallelograms, which are the key to the delta robot's functionality [2]. In our recent work [3], we presented a gripper based on two prismatic delta manipulators using 3D-printed parallelogram links presented in [4]. Unlike traditional parallel jaw grippers, our robots have compliant end-effectors, which makes them modular and accessible. This 6-DoF system is able to perform dexterous manipulation tasks, such as aligning a pile of coins, picking up a card from a deck, plucking a grape off of a stem, and rolling dough.

B. Robot hand and finger design for dexterous manipulation

Current robot hands with fingers span a range of different designs and complexity. Basic two-fingered grippers often have a single DoF, while complex anthropomorphic hands will often have multiple DoF per finger [5], [6]. Current designs use serial mechanisms for the individual fingers, similar to human hands. However, the more dexterous designs either require relatively bulky motors to be placed in the fingers, where they significantly increase the inertia, or they are actuated by cable drives [7], which are subject to highly non-linear effects and temporal variations due to slack and friction along the cables.

C. Dynamic surfaces

Dynamic surfaces have the potential to be used not only for object manipulation, but also as shape-changing interfaces. Distributed manipulation systems have many types, such as vibrating plates [8], actively controlled arrays of air jets [9], planar micromechanical actuator arrays [10], and actuated workbenches using magnetic forces [11]. These dynamic surfaces with an actuator array are also widely used in interactive displays. However, prior works focus on the motion on a plane, rather than working on the motion in 3D space. An additional DoF on the normal surface allows the delta array to interact with objects while utilizing contacts in 3D space.

D. Distributed and dexterous manipulation primitives

Dexterous manipulation using soft grippers is a challenging task due to stochasticity in the kinematics of the soft robot bodies. Various robust model-based control strategies use Lagrangian formulations to model the system [12] [13]. Another way is to use human demonstrations and Dynamic Motion Primitives (DMPs) to generate dynamically constrained trajectories for manipulation [14] [15] [16]. However, using analytical methods for a multi-agent system can lead to excessively high demand in compute, and generating human demonstrations for such a high degree of freedom system is logistically infeasible. Thus, we deploy a model-free RL algorithm directly on the hardware to generate trajectories for pushing and tilting an object against other robots to demonstrate the dexterous manipulation capabilities of the delta arrays.
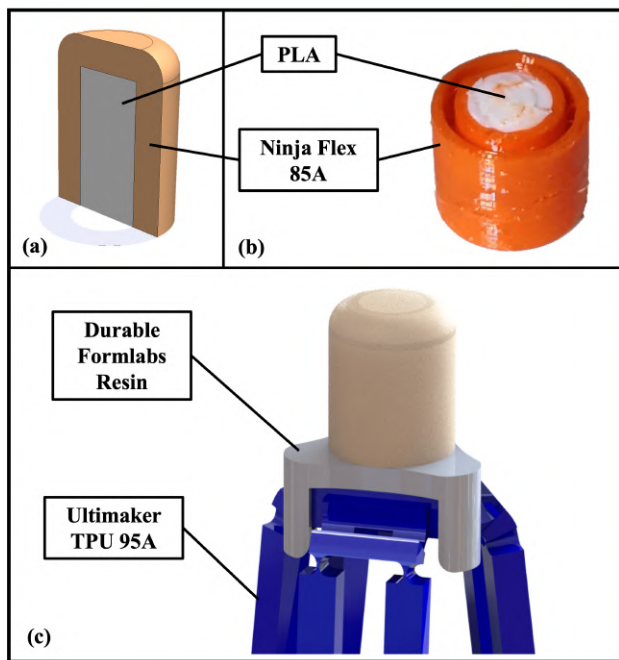
Fig. 3: (a) Cross-section view of the fingertip assembly in CAD (b) Cross-section view of 3D printed fingertip shows air pocket created by 0% infill (c) Attachment clip latches into underside of link using catch on end of the 3 prongs.

III. Prismatic Delta Robots

A delta array consists of multiple delta robots arranged in a planar grid structure. In this section, we explain the design of the individual delta robots. Each delta robot consists of three actuators connected by a parallel-bar linkage end-effector platform, as shown in Fig.2.

A. Actuators

Delta robots are often designed with rotational actuators that provide torque to individual links [2], [17]. These designs provide rapid and precise movements at the end-effector, but at the cost of a wide robot base. Due to the excessive width, which would conflict with the goal of creating a closely packed array of delta robots, we utilize linear actuators that enable us to position each robot in close proximity to one another. The linear actuators (Actuonix) have a 10 cm stroke length and internal potentiometers to provide analog feedback for position control. The end effector design is based on our previous work [3].

B. End Effector and Parallelogram Linkages

The end-effector platform is connected to three actuators through a parallelogram link, which converts linear motions into precise 3D x-y-z motions at the end-effector while keeping the platform parallel to the base.

The delta links are 3D printed as a single part with living hinges. Additional details can be found at [4]. The delta links were printed using thermoplastic polyurethane (TPU) for its low Young's modulus. This compliance allows the robots to safely interact with objects, other robots, and reduce the wear and tear of the system.

To perform dexterous manipulation, we design a fingertip that is inspired by the texture and feel of a human finger. The fingertip is attached as an end-effector to the delta platform using a reusable clip printed using durable formv3 resin for strength and flexibility. The fingertips were 3D printed using Polylactic acid (PLA) for the inner bone structure and NinjaFlex 85A for the outer skin which was fused together using a dual extrusion printer. Fig. 3 shows the cross-sectional structure of the design.We achieve this hollow structure by using 0% infill for the NinjaFlex and thin, two-layer walls which results in an enclosed cavity that provides the surface compliance for the finger.

C. Delta Robot Workspace

A key benefit of the prismatic delta design is that the workspace of the delta's end-effector extends beyond the triangular footprint of the three actuators. For our implementation, the horizontal distance between the centers of two actuators in a delta robot is 2cm, while the width of the workspace is approximately 6cm. To avoid excessive collisions between neighboring deltas, we restrict the horizontal workspace to a diameter of 3cm. The vertical workspace corresponds to the 10cm stroke length of the actuators.

The delta robots are operated within a workspace that is far away from their singularities. Ambiguities in the inverse kinematics can therefore be easily resolved to determine a suitable joint trajectory for a given desired end-effector trajectory.

IV. Modular Array Structure

Sets of delta robots are arranged into hexagonal grids to create delta arrays. Rather than constructing an array out of single deltas, we instead developed a modular 2×2 array unit for four deltas. Each unit can be operated in a standalone manner and provides a shared set of electronics and microcontrollers. To create an 8×8 array, we simply place 16 of the modules in a 4×4 macro grid, and a central computer then communicates to all of the modules to create coordinated manipulation strategies. The 2×2 modules thus provide a modular and extendable basis for easily constructing arrays of different sizes and replacing parts as needed. Our 8×8 configuration allows the manipulation of objects of a range of sizes and demonstrates the potential of such arrays in dexterous tasks.

A. 2×2 Delta Modules

Each 2×2 module employs a hexagonal structure as shown in the middle image of Fig.2. The linear actuator bodies are held together using two laser-cut plexiglass plates, which are, in turn, supported by aluminum stand-offs. The stand-off configuration equally compresses the 12 linear actuators from both sides forming a stable structure.

Each robot in the module is then secured through the base of the linear actuators using a 3D-printed connector made from PLA and then attached to a 3D-printed enclosure made of PLA. This hardware box houses the electronics needed to control the four deltas in that module and allows the module
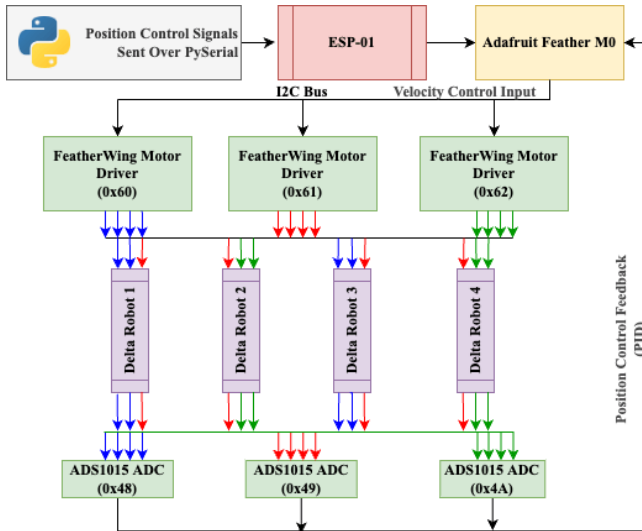
Fig. 4: Communication flowchart for a 2×2 module. The control of three actuators of each robot in a four-robot module is accomplished by three motor drivers. Colored arrows show the distributed control framework between drivers, actuators, and ADCs.
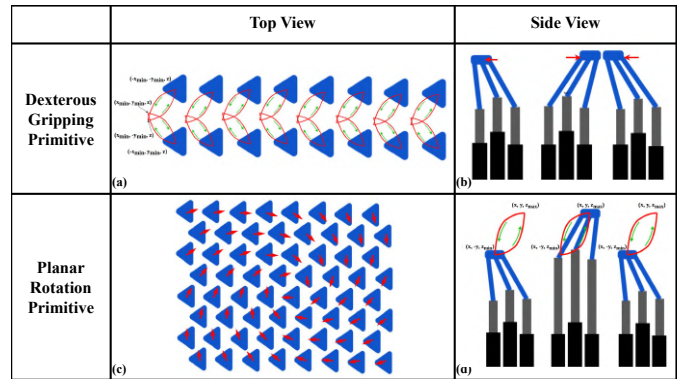

Fig. 5: Top-view and side-view representation of the ellipsoid trajectories of the two-beat gaits. (a) represents 2 odd rows from the array for execution of the Dexterous Gripping Primitive and (c) represents the entire delta array executing the Planar Rotation Primitive. Both follow two-beat gait patterns shown in (b) and (d) respectively

to be connected with a base plate that supports the array. The resulting 2×2 delta modules offer a balance between modularity and ease of maintenance.

For the face-down mode of operation, the setup is inverted and mounted onto pillars made of 80/20 aluminum extrusions—constrained at the top by the base plate and on the bottom by an optical breadboard. Between the array and the breadboard, a clear plexiglass platform is supported by a movable linear slide on each corner. Planar manipulations are performed ontop of this platform. These sliders allow the height of the platform to be manually adjusted depending on the size of the objects being manipulated. The transparent platform enables our vision pipeline to track the pose of objects from below as they are manipulated.

B. Electronics

Adafruit Feather M0 boards control four deltas (12 actuators) in a module and they are housed in the hardware enclosure box. Adafruit DC Motor/Stepper FeatherWing is used to control the velocity of the end effectors through PWM signals. We use an analog-to-digital converter (ADC) to couple the position feedback from the linear actuators. This also acts as a low-pass filter to eliminate high-frequency noise from the electromagnetic interference generated in the circuit for precise position control. We use a 12-bit ADC that resolves the 100 mm length, and we apply a low level PID control with a final precision of upto 0.3 mm.

We use the I^2C bus on the Feather M0 and distribute the data bus, clock, and a 12W power supply across three FeatherWing motor drivers using a custom electronic shield circuit. The compactness of the design allows us to maintain close proximity among all the delta modules.

For perception, we mount a USB camera module on the bottom plane looking upwards. The camera tracks object poses with minimal occlusions while dexterous manipulation

is being performed by the delta array from above.

C. Communication Across the Array

To efficiently control the entire array of 64 robots, communication factors like latency, noise, and amplitude of signal need to have stable optimal values. Instead of using TTL communication using wires, which results in exhaustive cable management and noise, we use off-the-shelf ESP-01 WiFi modules operating at 115,200 baud rate, enabling low latency, low noise, and speedy wireless communication. A high-level flowchart of communication is shown in (Fig. 4). We use protocol buffers (protobuf) to transmit control data because of their high compressibility and effectiveness in networked communications as shown in [18].

V. PREDEFINED DISTRIBUTED MANIPULATION STRATEGIES

The 8×8 array in "facing-up" mode can execute a variety of dexterous manipulation strategies distributed across its delta robots. These strategies include planar manipulations like translation, rotation, and convergence, as well as out-of-plane and prehensile manipulations. To test the capabilities of the delta array, we implemented a series of basic manipulation policies. Each delta robot is given (x, y) coordinates representing its position $\vec{p}$. For every primitive, we use simple linear algebraic operations to determine the position-controlled trajectories of the delta robots.

The delta array policies are designed as two-beat finger-gaiting strategies that repeatedly cause the deltas to make and break contact with the objects being manipulated. The planar trajectory moves in the vertical direction, with a constant gait, and along a horizontal direction as given by a high-level primitive. The movements can be considered as going from $[-\vec{p}, z_{min}]$ to $[\vec{p}, z_{max}]$ as shown in Fig.5A, where $z_{min} = 7cm$ and $z_{max} = 10cm$ are the alternating end-effector positions on $Z - axis$. The two-beat gait means that half of the deltas in the array will be in an up configuration while the other half are in a down configuration, i.e., 180 degrees out of phase. We use a two-beat gait to maximize
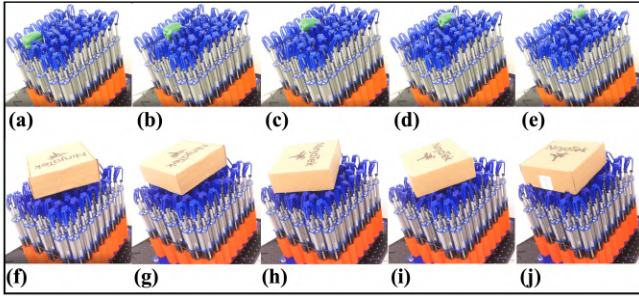
Fig. 6: The rows of images demonstrate manipulations of different objects using the 8×8 delta array. The numbers beneath each row indicate the timestamp. (A) A toy bell pepper object that weighs $4g$ with a characteristic length of $60mm$ is transported from one edge of the array to the other using a dexterous gripping primitive from (a) to (e). (B)A box object is rotated while the position on the array stayed the same as shown from (f) to (j).

Fig. 7: Using the delta robots for tilting an object using learned trajectories

the number of deltas in contact with the object at a given time as described in [19].

A. Dexterous Gripping Primitive

Apart from purely planar manipulation strategies, we present a *grip-and-push* primitive that can be deployed to grasp objects within a line of delta robots and push the object forward or backward along the line. We use a two-beat finger gait with a constant Z value, and periodically switching Y_{max} and Y_{min} to push objects along the $X-axis$. A demonstration of the strategy on a foam bell pepper is shown in Fig.5[(a), (b)] and Fig. 6A.

B. Planar Translation Primitive

For planar translations, a straightforward implementation of up, down, left, and right movements can be shown by placing a point along the X and Y axes at infinity, computing the unit distance vector from the center of the robot to that point and plug the unit vector in the aforementioned two-beat finger-gaiting pattern for planar translation of objects on the surface of the linear delta arrays.

C. Planar Rotation Primitive

For planar rotation, the distance vector from each robot to the center is multiplied by the rotation matrix to generate rotating unit vectors for planar rotation as shown in Fig.5[(c), (d)] and Fig.6B.

D. Wall Primitive

A unique feature of linear delta arrays is the ability to use a subset of delta robots to form walls of various shapes for restricting the movements of objects. Dexterous tasks like clamping or aligning an object along the wall and turning it around for inspection can be performed using simple yet effective policies, an inverted version of which we present in the next section.
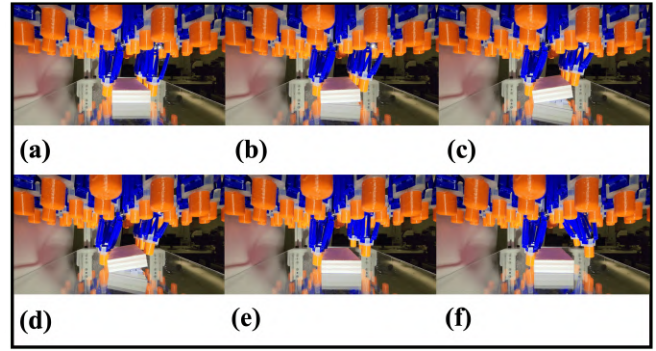
VI. LEARNING DEXTEROUS MANIPULATION STRATEGIES

The array can also be used to learn basic dexterous manipulation skills. We design an RL environment on the real hardware with the delta array in the facing-down mode 7. From the robot's camera, we track the SE(3) pose of a checkerboard pattern attached to a wooden block that the robot should manipulate using the neighboring 6 delta robots. The pose of the object is tracked to generate the reward for performing the task. We compute the L2 error between the current pose and desired pose and compare it with a threshold of 0.1 cm for translation and 0.5 rad for rotation. We use the following formula to compute the reward to maximize:

$$f(x) = \begin{cases} -1 * T_e - 3 * R_e, & \text{if } T_e > 0.1 \vee R_e > 0.5 \\ +10, & \text{otherwise} \end{cases} \quad (1)$$

Where T_e is the translational error and R_e is the rotational error.

The errors generated by the vision system are used to train fingertip trajectories for grasping and tilting the wooden block using episodic relative entropy policy search (eREPS) [20]. The trajectories are generated by weighting the output of eREPS on 5 basis functions of the DMPs. eREPS is a model-free RL algorithm that iteratively optimizes a Gaussian distribution over the weights of the DMP. We initialize the distribution with a zero mean and a diagonal covariance matrix of 0.7.

In each execution episode, the robot samples the weights of the DMP, generates and executes the corresponding fingertip trajectory, and computes the resulting reward. The Gaussian policy is updated every ten episodes. The algorithm runs until a success rate of at least 90% has been achieved, which takes approximately 270 epochs.

VII. EXPERIMENTS

This section describes the experimental evaluations performed using the delta array.

A. Facing-up Manipulation Experiment

We constructed an 8×8 delta array using the design described in Sections III and IV. We implemented the distributed manipulation strategies explained in Section V
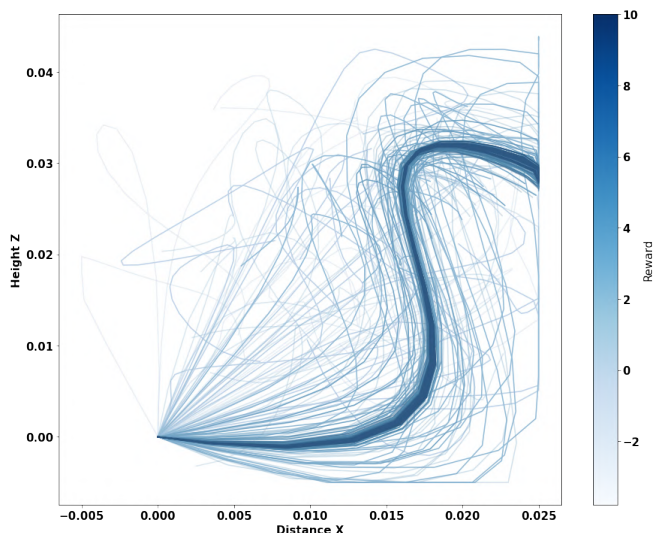
10328

Fig. 8: Using REPS to generate weights of the basis functions of a DMP to generate a 2D trajectory (curved lines from (0,0) till endpoint predicted by REPS) for the delta robots. Faint trajectories represent initial exploration trajectories with low/negative rewards, darker trajectories represent learned exploitation with high rewards

for the upward facing configuration. The robot could then manipulate objects placed on top of the array. The robot successfully performed non-prehensile translation and rotation manipulations of objects of dimensions ranging from 60mm × 40mm × 20 mm to 300mm × 300mm × 80mm and weights ranging from 4g to 1kg. Examples of different manipulations are shown in Fig. 6. Each picture represents a snapshot of the manipulation task being performed. Our methods work on objects with a stable center of mass and a moderate coefficient of friction. Objects like soda cans with a shifting center of mass are hard to manipulate in an open-loop control setting.

B. Facing-down Manipulation Experiment

We implemented the learning of dexterous manipulation strategies in Section VI for the downward facing configuration. This was done only in the facing-down configuration due to the ease of resetting the environment as compared to the upright configuration. The robot could learn to grasp and tilt objects placed on a planar surface directly in the real world without simulation. The face-down mode provides a more stable environment for easier resetting of objects between episodes. Throughout the training, the trajectories generated by the robot are shown in Fig. 8.

In the initial stages, the algorithm explores the action spaces while generating very low rewards shown as the faint tinted trajectories in Fig. 8. As the training progresses, the actions converge to more optimal values and the model becomes exploitative to generate maximum reward consistently towards the end of the training as shown by the darker lines in Fig. 8. The learning approach could be used to generate DMPs for other shapes of objects as well. Due to the soft linkages, heavy objects with smooth surfaces are hard to lift using the delta robots.

C. Discussion

For the upward facing configuration, the results show that the delta arrays can be used for a variety of manipulation types. In-plane translation and rotations perform better when applied to larger objects where more delta robots make contact with the objects at any time. In some cases, smaller objects can get stuck between deltas in the array, but the compliance keeps the system safe in these situations. The weight of objects plays an important role in the performance of the non-prehensile manipulations as well. We found that heavier objects tended to be manipulated more easily. Part of this may be due to the correlation in size and higher friction forces between the delta robots and the object.

The wall policy allows the delta array to successfully align objects against the side of sets of delta robots. In this manner, the delta array can remove some of the uncertainty of the object's position. The wall policy can also be seen as a hybrid policy that combines the use of the translational policy with using some fingers as fixtures/obstacles. The delta array thus presents a suitable base for exploring a variety of mixed manipulation strategies in the future.

For the downward facing configuration, we observe that the design of the fingertips played an important role since higher friction generated by the fingertip surface made objects easier to tilt against the walled fingertips. The hollow cavity in the fingertips helps the surface to conform to the surface of the object to make lifting the objects easier. The compliance of the robots also adds robustness to the policy which makes it slightly more sample efficient.

VIII. CONCLUSION

We proposed delta arrays as a new type of dexterous manipulation robot. We presented the assembly of individual delta robot modules for close packing and overlapping workspaces, and we proposed a modular design and distributed control framework. We constructed and tested an 8×8 array of delta robots and showed that the delta arrays can be used to perform a variety of manipulation strategies in two different configurations. For the facing-up configuration, our current manipulation primitives tend to be better suited for larger objects, where the redundancy of the array provides robustness. While for the facing-down configuration, we were able to learn grasping and tilting strategies directly on the hardware with low-level visual supervision.

To show generalizability across objects, we plan to incorporate vision feedback to extract object poses and learn grasping strategies to generalize to objects of various shapes and sizes. We also plan to extend the use of the delta arrays for deformable object manipulation and to demonstrate the effectiveness of compliant multi-agent manipulation systems.

IX. ACKNOWLEDGEMENTS

REFERENCES

[1] R. Clavel, "Device for the movement and positioning of an element in space," Dec. 11 1990, uS Patent 4,976,582.

[2] L. Rey and R. Clavel, "The delta parallel robot," pp. 401–417, 1999.

[3] P. Mannam, A. Rudich, K. Zhang, M. Veloso, O. Kroemer, and F. Temel, "A low-cost compliant gripper using cooperative mini-delta robots for dexterous manipulation," *Robotics: Science and Systems XVII*, 2021.

[4] P. Mannam, O. Kroemer, and F. Z. Temel, "Characterization of Compliant Parallelogram Links for 3D-Printed Delta Manipulators," in *Experimental Robotics*, B. Siciliano, C. Laschi, and O. Khatib, Eds. Cham: Springer International Publishing, 2021, vol. 19, pp. 75–84, series Title: Springer Proceedings in Advanced Robotics. [Online]. Available: https://link.springer.com/10.1007/978-3-030-71151-1_7

[5] U. Kim, D. Jung, H. Jeong, J. Park, H.-M. Jung, J. Cheong, H. R. Choi, H. Do, and C. Park, "Integrated linkage-driven dexterous anthropomorphic robotic hand," *Nature Communications*, vol. 12, no. 1, p. 7177, Dec. 2021. [Online]. Available: https://www.nature.com/articles/s41467-021-27261-0

[6] O. M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020. [Online]. Available: https://doi.org/10.1177/0278364919887447

[7] O. M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[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, 1995, pp. 1989–1996 vol.2.

[9] M. Yim, J. Reich, and A. A. Berlin, "Two approaches to distributed manipulation," in *Distributed Manipulation*. Springer, 2000, pp. 237–261.

[10] G. Bourbon, P. Minotti, P. Htlin, and H. Fujita, "Toward smart surfaces using high-density arrays of silicon-based mechanical oscillators," *Journal of intelligent material systems and structures*, vol. 10, no. 7, pp. 534–540, 1999.

[11] G. Pangaro, D. Maynes-Aminzade, and H. Ishii, "The actuated workbench: Computer-controlled actuation in tabletop tangible interfaces," in *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '02. New York, NY, USA: Association for Computing Machinery, 2002, p. 181–190. [Online]. Available: https://doi.org/10.1145/571985.572011

[12] A. Kazemipour, O. Fischer, Y. Toshimitsu, K. W. Wong, and R. K. Katzschmann, "A robust adaptive approach to dynamic control of soft continuum manipulators," *CoRR*, vol. abs/2109.11388, 2021. [Online]. Available: https://arxiv.org/abs/2109.11388

[13] G. Zhang and D. Cheng, "Adaptive fault-tolerant guaranteed performance control for euler-lagrange systems with its application to a 2-link robotic manipulator," *IEEE Access*, vol. 8, pp. 184 160–184 171, 2020.

[14] J.-P. Saut, A. Sahbani, S. El-Khoury, and V. Perdereau, "Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 2907–2912.

[15] R. Jäkel, S. R. Schmidt-Rohr, S. W. Rühl, A. Kasper, Z. Xue, and R. Dillmann, "Learning of Planning Models for Dexterous Manipulation Based on Human Demonstrations," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 437–448, Nov. 2012. [Online]. Available: https://doi.org/10.1007/s12369-012-0162-y

[16] A. Gupta, C. Eppner, S. Levine, and P. Abbeel, "Learning dexterous manipulation for a soft robotic hand from human demonstrations," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 3786–3793.

[17] Sam Alvares, "Building DeltaZ," Aug. 2021. [Online]. Available: https://www.youtube.com/watch?v=Mg2i0toDOXg

[18] S. Popić, D. Pezer, B. Mrazovac, and N. Teslić, "Performance evaluation of using protocol buffers in the internet of things communication," in *2016 International Conference on Smart Systems and Technologies (SST)*, 2016, pp. 261–265.

[19] S. Thompson, P. Mannam, Z. Temel, and O. Kroemer, "Towards robust planar translations using delta-manipulator arrays," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6563–6569.

[20] J. Peters, K. Mülling, and Y. Altün, "Relative entropy policy search," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, ser. AAAI'10. AAAI Press, 2010, p. 1607–1612.