Co-Designing Manipulation Systems Using Task-Relevant Constraints

Apoorv Vaish[1] Oliver Brock[1,2]

***Abstract*— A robotic system's hardware and control policy must be co-optimized to ensure they complement each other to interact robustly with the environment. However, this combined search is extremely high-dimensional and intractable without a suitable underlying representation. This paper uses *environmental constraints* to structure the co-design space for manipulation. We show that task-relevant constraints encode regions of the search space containing reasonable co-design solutions. Furthermore, this underlying representation renders a co-design space amenable to gradient-based optimization. For efficient search, we present the *co-design Jacobian* that describes how the robot's motion varies with control as well as hardware design changes. This Jacobian exploits the structure induced by environmental constraints for iterative design updates in the co-design space. Using these two conceptual tools, we co-design manipulators, grippers, and multi-fingered hands, showing that environmental constraints are an effective representation for co-designing diverse manipulation systems. Our methodology also scales well with increased co-design parameters, rendering the co-design of complex, high-dimensional manipulation systems feasible.**

I. Introduction

The capabilities of a robotic system depend on how well the hardware and the control policy complement each other. Therefore, we must co-design hardware and control strategies to ensure synergistic performance. However, the search space spanned by hardware and control parameters is high-dimensional, rendering the co-design problem extremely challenging. Furthermore, we still lack established practices that make co-design effective for complex robotic systems. Therefore, outlining fundamental principles that make this high-dimensional search tractable will help us design more competent robots. We investigate this in the context of robot manipulation involving contact.

We present two conceptual tools for co-designing complex manipulation systems. Firstly, we propose to structure the high-dimensional design space, spanned by the hardware and control parameters, using the notion of *environmental constraints* [1]. Environmental constraints (ECs) implicitly capture the combined effect of the pertinent hardware and control parameters, thereby encoding the task-relevant features of the search space. Furthermore, we show that an underlying representation based on ECs produces a co-design space amenable to gradient-based optimization.

[1] Robotics and Biology Laboratory, Technische Universität Berlin
[2] Science of Intelligence (SCIoI), Cluster of Excellence, Berlin, Germany
We gratefully acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2002/1 "Science of Intelligence" and the German Priority Program DFG-SPP 2100 "Soft Material Robotic Systems".

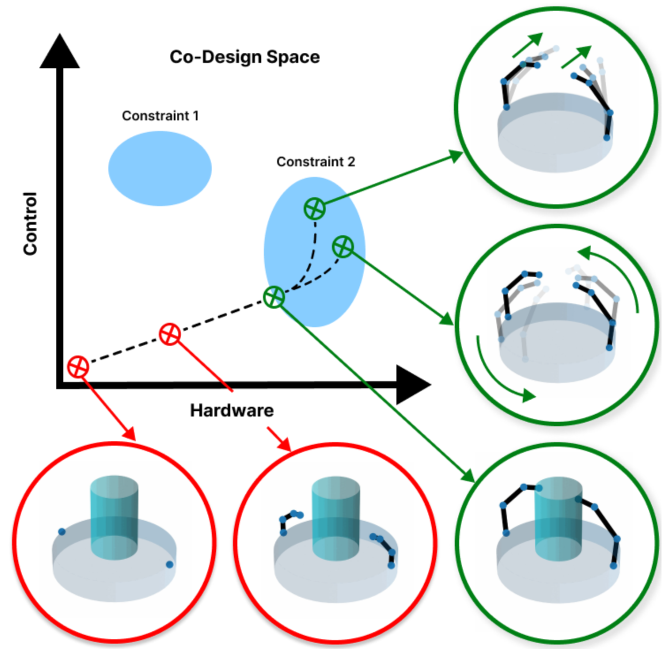

Fig. 1. High-dimensional co-design of manipulation systems is feasible using task-relevant constraints. A randomly initialized point (⊗) in the co-design space, spanned by hardware and control parameters, is directed to the blue regions containing competent co-designs (⊗) that form task-relevant constraints by gradient-ascent (dashed lines) using the co-design Jacobian.

The second tool we present is the *co-design Jacobian* that exploits the structure induced by ECs. While the conventional robot Jacobian describes how the robot's motion varies with changes in the control inputs, the co-design Jacobian augments this with hardware parameters, capturing how the robot's motion varies with changes in hardware design and control inputs. Together with the structure induced in the search space by ECs, the co-design Jacobian is a conceptually intuitive and computationally efficient tool for efficient gradient-based optimization in the co-design space.

We use these tools to co-design diverse manipulation systems, like robot arms, grippers, and multi-fingered hands. Our experiments demonstrate efficient co-design in 80-dimensional spaces, taking less than 1.5 minutes on a single 3.50GHz CPU. The tools described here scale well in computational complexity when we increase the dimensionality of the co-design space to as high as 2560, more than enough for practical robotic systems. Our results demonstrate that ECs provide insights into the structure of the co-design space, rendering the co-design of complex, high-dimensional manipulation systems feasible.

II. Related Work

We first discuss the need for task-specific priors like environmental constraints (ECs) that impose an exploitable structure on the high-dimensional search space, enabling efficient co-design. We also review various optimization methods and motivate a co-design approach that is conceptually intuitive and computationally efficient.

A. Environmental Constraint Exploitation

Co-design requires a search space spanned by hardware design and control parameters. Researchers have explored various design representations like primitive shapes [3], parametrized meshes [4], design graphs [5], voxel grids [6], cage-based deformations [7], and relied on a spectrum of physics-based and data-driven models for control. These design and control representations define an extremely high-dimensional search space without encoding task-specific information. For efficient search, a suitable representation must also encode the task-relevant features of the design space, guiding the search directly to pertinent co-designs. Task-specific priors like gaits and foot contact timings were useful for animating animal locomotion [8] as well as co-designing legged robots [9], [10]. These priors render the co-design of legged locomotion effective by imposing an exploitable structure to the search space. However, no such candidate offers insights into the structure of co-design space for manipulation.

We propose using environmental constraints (ECs) to structure the co-design space for manipulation systems. EC exploitation refers to processes that leverage the environment to reduce uncertainty in state variables [1]. It has proven to be an effective tool for designing both hardware [19] and control strategies [20], [21] for manipulation. We argue that task-relevant ECs capture the combined effect of pertinent hardware and control and their interaction with the environment. Therefore, an underlying representation based on ECs encodes the regions of the co-design space relevant to manipulation tasks.

B. Optimization Methods in Co-design

Researchers have employed various optimization methods for co-designing synthetic agents. Evolutionary algorithms like particle swarm optimization [4] and evolutionary strategies [6] directly search for globally optimal co-designs. Augmented with gradients, they emulate the intertwined process of evolution and learning through a nested optimization: the outer loop samples a population of hardware designs, and the inner loop asynchronously optimizes a neural network-based control policy [3], [11], [12]. However, this approach decouples the functional relationship between hardware and control. The notion of a "hardware policy" maintains this relationship by simultaneously propagating gradients concerning both hardware and control parameters through the computational graph [13]. These gradient-based co-design approaches are computationally more efficient than gradient-free optimizations [13], [7], but only return locally optimal co-designs.

Differentiable physics simulators compute gradients concerning various simulation parameters. There are differentiable simulators developed for co-designing soft robot locomotion [14], locomotion in diverse environments [15], and aerial robots [5]. Simulators built for contact-rich manipulation achieve competent co-designs in fewer episodes than evolutionary strategies or reinforcement learning [7]. However, differentiable simulators require significant software development resources, and differentiable contacts are tricky to model [16]. We aim for an intuitive and efficient co-design approach using the proposed *co-design Jacobian*. Similar methods that co-optimize design and trajectory parameters rely on an augmented Jacobian [17], [18]. However, with the structure induced by ECs, the co-design Jacobian enables the co-design of manipulation systems for contact-rich tasks.

III. Constraint-Guided Co-Design

We propose structuring the co-design space using Environmental constraints (ECs). To effectively impose this structure, we evaluate the task performance of a constraint generated by a co-design. To efficiently leverage this structure, we compute the gradients of this performance for iterative design updates in the co-design space. Since evaluating the gradients is critical, we first outline a template for formulating co-design problems suitable for gradient-based optimization. We then use this template to formulate gradient-based co-designs of manipulation systems in search spaces structured by ECs.

A. Template for Gradient-Based Co-Design

We present a template outlining the algorithmic components of co-designs suitable for gradient-based optimization:

1) Search Space: The search space, S is an $(m+n)$-dimensional continuous space spanned by control parameters, $(\theta)_m$ and hardware parameters, $(h)_n$.

2) Quality Function: The task imposes a structure on S that a quality function, f, must quantify. f should ideally render a smooth optimization landscape depicting the task relevance of co-designs in S, forming peaks in the task-relevant regions while penalizing task-irrelevant ones. f can also incorporate costs associated with co-designs in S.

3) Optimization Method: Gradient-based methods are ideal for navigating the landscape rendered by f in high-dimensional, continuous search spaces like S. However, f must be differentiable for evaluating gradients, $\left(\frac{\partial f}{\partial \theta_i}, \frac{\partial f}{\partial h_i}\right)$. Furthermore, we can only reach the local maxima by iteratively updating θ_i and h_i from a point in S.

B. Constraint-Guided, Gradient-Based Co-Design

We describe our methodology using the algorithmic components discussed in section III-A. First, we qualitatively describe how ECs structure the co-design space, S. We evaluate the task relevance of a constraint using a candidate quality function, f. Finally, we present the co-design Jacobian that exploits the structure induced by ECs to reach locally optimal co-designs in the search space.

1) Search Space: ECs form patches in S as shown in Fig. 1. Their boundary separates the manipulation-relevant co-designs that interact with the environment (blue regions) from co-designs that do not.

2) Quality Function: Ideally, f must generate a smooth optimization landscape over S for its gradients to achieve two purposes: a) Outside the EC patches in S, they guide the search to a local EC patch. b) Within an EC patch, they lead to co-designs forming constraints better suited to the task. To define f, we use the forward kinematic equations describing the end-effector pose, x, as a function of co-design parameters, q_c. We show that the inverse kinematics equation of a redundant manipulator [22], described in Eqn. 1, encodes the two purposes.

$$\begin{aligned} x &= f(q_c) \\ \partial q_c &= J_c^{+}\partial x + \left(I - J_c^{+} J_c\right)\kappa \end{aligned} \tag{1}$$

Here, x is the current Cartesian pose of the end-effector, and ∂x is the error from the required end-effector pose. J_c is the co-design Jacobian described in the next section, while J_c^{+} is its pseudoinverse. As we will show, the former term $J_c^{+}\partial x$ directs the search towards the constraint patches in S. The latter term $(I - J_c^{+} J_c)\,\kappa$ corresponds to internal changes in q_c that do not result in end-effector motion. We choose $\kappa = \alpha \frac{\partial T}{\partial q_c}$, where α is a constant, and T is the the task-compatibility index [22] described in Eqn. (2).

$$T = \sum_{i=1}^{l} w_i \underbrace{\left[u_i^{\mathrm{T}}\left(JJ^{\mathrm{T}}\right)u_i\right]^{\pm 1}}_{\text{Force}} + \sum_{j=l+1}^{m} w_j \underbrace{\left[u_j^{\mathrm{T}}\left(JJ^{\mathrm{T}}\right)^{-1}u_j\right]^{\pm 1}}_{\text{Velocity}} \tag{2}$$

Here, J is the robot Jacobian. $\mathrm{u_i}$, $\mathrm{u_j}$ are the directions of interest for force and velocity transmission. A $+$ sign is used when the resolution of force/velocity transmission is of interest, whereas a $-$ sign is used when the magnitude is of interest. $\mathrm{w_i}$, $\mathrm{w_j}$ weigh the relative importance of m tasks. This index measures the robot's ability to generate constraints that transmit forces and velocities in the directions dictated by the task. Thus, the latter term in Eqn. (1) adjusts the co-design parameters to form constraints better suited to the task.

3) Optimization Method: The conventional robot Jacobian, J, captures how changes in control inputs, θ_i lead to changes in robot motion, x. For co-design, we formulate an augmented *co-design Jacobian*, $\mathrm{J_c}$ that captures how changes in either control, θ_i or hardware, h_i parameters lead to changes in robot motion, x. J_c exploits the structure induced by ECs to improve the co-design parameters, $q_c = \{h_i, \theta_i\}$. J_c is conceptually similar to the conventional robot Jacobian, making co-design conceptually intuitive and approachable to roboticists. Eqn. (3) describes both the Jacobians.

$$J = \left[\frac{\partial x}{\partial \theta}\right],\; J_c = \left[\frac{\partial x}{\partial \theta}, \frac{\partial x}{\partial h}\right] \tag{3}$$

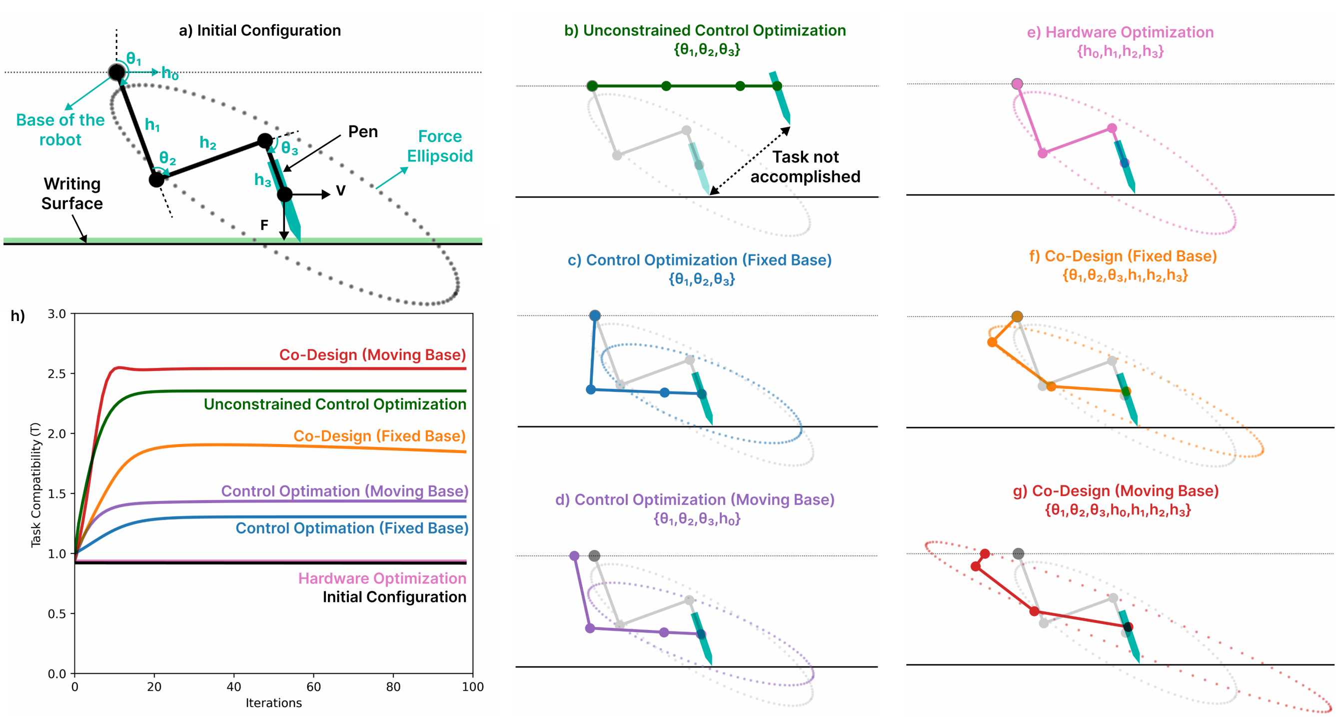

Fig. 2. A 7-dimensional co-design of a manipulator guided by the constraint formed during the task of writing on a horizontal surface using a pen at its end-effector. a) The initial arm configuration. The manipulator requires a high vertical force resolution (F) and horizontal velocity resolution (V) for the best writing performance. The final co-designs and their force ellipsoids are shown for different optimization scenarios in b-g. b) Control parameters optimized without task constraints lead to a robot configuration that doesn't fulfill the task requirements. c) Control optimization with a fixed robot base. d) Control optimization with an additional hardware parameter that moves the robot base. e) Hardware optimization. f) Co-designing hardware and control. g) Co-designing hardware and control with an additional parameter that moves the robot base. h) Task-compatibility progression of co-designs in b-g.

IV. Experimentation

We first show that environmental constraints (ECs) encode the regions of the co-design space relevant to manipulation. To demonstrate this, we compare the performance of robots designed with and without the insights of task-relevant constraints. In the process, we show that the co-design Jacobian is a simple but effective tool that exploits the structure imposed by ECs to reach competent co-designs. We also study how random initialization affects the ensuing gradient-based co-design. Finally, we evaluate how our methodology scales for complex, high-dimensional co-designs.

A. Redundant Manipulator Co-Design

In this experiment, we co-design a 3-dof redundant manipulator to write on a horizontal surface using a pen at its end-effector, as shown in Fig. 2a. As we will show, without the insight of task-relevant EC, the optimization leads to a robot configuration that does not fulfill the task. We then leverage the task-relevant EC to co-design multiple manipulators.

The optimization intitializes with hardware and control parameters taken from [22], and shown in Fig. 2a: link lengths, $h_i = [0, 1, 1, 0.5]$, and joint angles, $\theta_i = [160°, 90°, -90°]$. The total link length, $(h_1 + h2 + h3)$ is maintained to be 2.5 throughout all optimizations. To co-design a robot forming a constraint suitable for the task of writing, we choose $m = 2$ for two tasks: high force resolution in the vertical direction and a high-velocity resolution in the horizontal direction, depicted by F and V in Fig. 2a. The corresponding directions of interest are $u_1 = [0,1]^{\mathrm{T}}$, $u_2 = [1,0]^{\mathrm{T}}$. The relative weights, $w_1 = 0.8$, and $w_2 = 0.2$, are chosen by trial and error. The co-design parameters q_c are updated for 100 iterations with $\alpha = 0.1$ in all experiments. The progression of task compatibilities, T, is shown in Fig. 2h.

First, we optimize co-design parameters without the task-relevant EC. We update the control parameters of manipulator, θ_i using the gradients, $\frac{\partial T}{\partial \theta_i}$ computed using Eqn. (2). The resulting arm pose is shown in Fig. 2b. While T increases to 2.29, the pen doesn't make contact with the surface, and the manipulator fails to form a task-relevant constraint. Thus, the manipulator will not be effective at writing, and the task is unaccomplished. For further experiments, we impose the task-relevant constraint during co-design. The manipulator forms a constraint against the surface in its initial configuration. To maintain this constraint, we must search within the null space of the manipulator. We substitute $\kappa = \alpha \frac{\partial T}{\partial q_c}$ in Eqn. (1) to compute the new updates, ∂q_c.

We use these ∂q_c for 5 different optimizations using different subsets of the 7 co-design parameters and show the results in Fig. 2c-g. First, we consider only the control parameters $\{\theta_i\}$ corresponding to the three joint angles. The resulting posture resembles a human arm while writing [22], as shown in Fig. 2c, and has $T = 1.26$. Optimization with an additional hardware parameter, h_0, that translates the robot base horizontally improves T to 1.34 as shown in Fig. 2d. In Fig. 2e, we only optimize the hardware parameters h_i. This results in no change in robot configuration, as there is no way to maintain the end-effector configuration with fixed control parameters. In Fig. 2f, we co-design using 3 control, $\{\theta_1, \theta_2, \theta_3\}$ and 3 hardware, $\{h_1, h_2, h_3\}$ parameters which improves T to 1.82. In Fig. 2g, we co-design using all 7 parameters, $\{\theta_i\}$ and $\{h_i\}$. T improves significantly to 2.56, showing that co-optimizing all parameters using the task-relevant constraints yields the best result compared to optimization limited to either hardware or control domains.

B. Gripper Co-design

We co-design a gripper with two key differences from the previous section. Firstly, the initial co-design parameters do not form a task-relevant constraint. Thus, the co-design Jacobian must lead the search towards regions of the co-design space containing designs making task-relevant constraints. Secondly, as we will show, a constraint optimized through a high-dimensional co-design represents a family of task-relevant co-designs differing in performance and control complexities. We introduce an informed method to guide this discrete search. Finally, we evaluate the effect of random initialization on our gradient-based co-design approach.

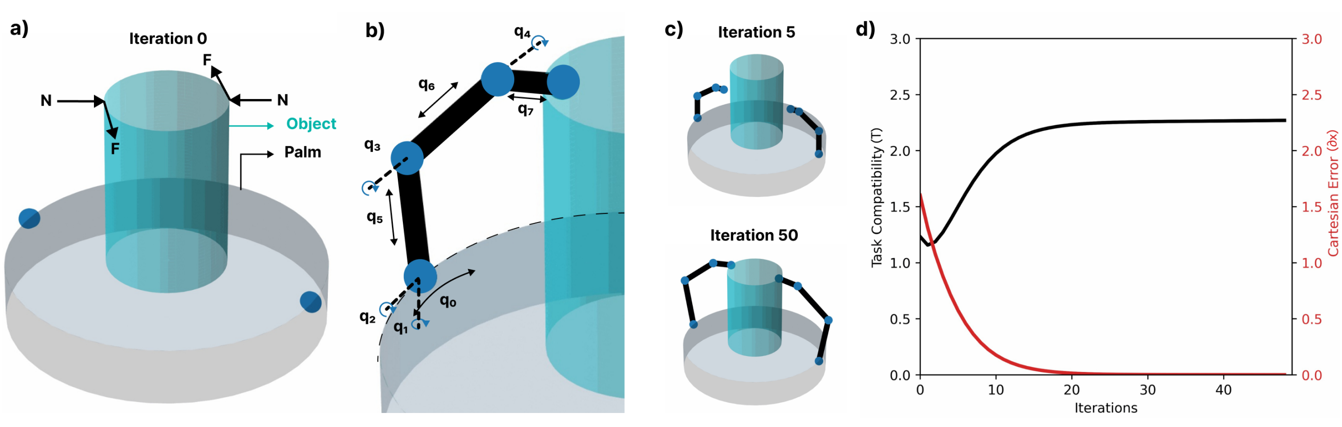

Fig. 3. The co-design of a two-fingered gripper shows that task-relevant constraints structure the co-design space, and the co-design Jacobian exploits this structure to reach competent designs. a) Each finger must form a constraint capable of applying forces in the shown directions, N and F, at two points to rotate the blue object on its palm. b) Each finger has eight co-design parameters forming the shown kinematic chain. c) Gripper design progression. d) Progression of task compatibility, T and the Cartesian error, ∂x during the optimization.

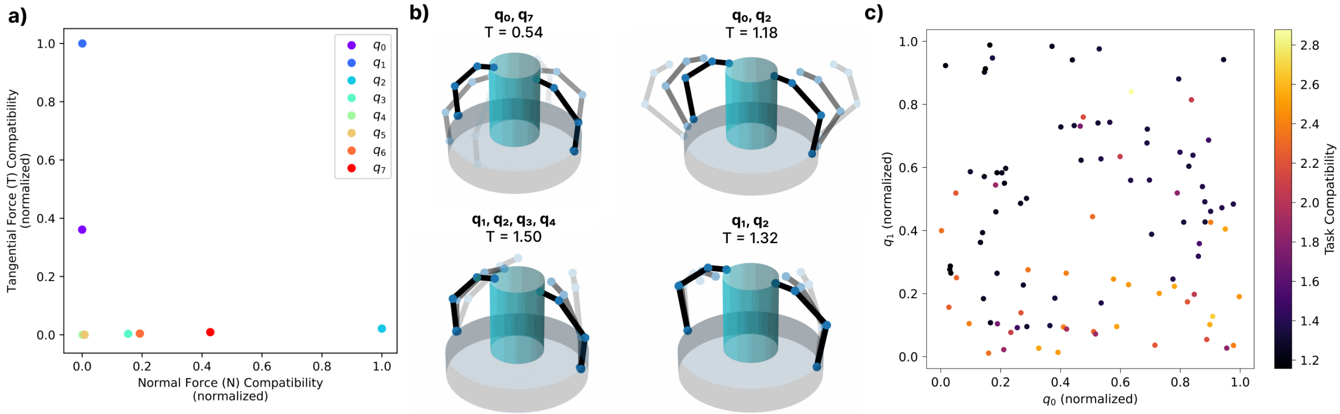

Fig. 4. Discrete co-design of grippers by fixing a subset of co-design parameters to form the fixed hardware, while the rest are controllable during the task. a) The individual task-compatibility contribution of each co-design parameter. b) We design four grippers by selecting a subset of the 8 co-design parameters that can be controlled during the task to render task compatibilities, T. There is a trade-off between the resulting performance and the rendered control complexity. c) Effect of initialization of parameters in the co-design space on the resulting task compatibility. Our approach is susceptible to seeking out local maxima, making reasonable initialization essential to ensure competent results.

1) Constraint Formation: In this section, we co-design a gripper to rotate an object in its palm. We require two fingers forming constraints adept at applying normal forces for grasping and tangential forces to rotate the object at two points along the top edge of the object, depicted by N and F in Fig. 3a. We update the co-design parameters using Eq. (2) but add squared parsimonious penalties for small link lengths in f. Two kinematic chains with different a_0 are initialized using the Denavit-Hartenberg parameters described in Table. I and shown graphically in Fig. 3b. The co-design parameters, q_i are initialized to $(0, 0, 90°, -45°, -45°, 0, 0, 0)$. The Cartesian error in this configuration, ∂x, is not zero; therefore, the object does not constrain the robot. On updating the co-design parameters using Eq. 1, the term $J_c^+ \partial x$ guides search to a design forming task-relevant constraint. The latter term adjusts the finger design to apply forces in directions dictated by the task. Fig. 3c shows the design progression during the 50 optimization steps. The corresponding task compatibility measure, T, and Cartesian error, ∂x, are plotted in Fig. 3d. This experiment shows that on initializing at a point outside the (blue) constraint patches described in Fig. 1, the co-design Jacobian exploits the structure induced by task-relevant constraints to reach locally optimal co-designs.

TABLE I
DENAVIT-HARTENBERG PARAMETERS FOR GRIPPER CO-DESIGN

i	a_i	θ_i	d_i	α_i
0	± 1	q_0	0	0
1	0	q_1	0	$90°$
2	q_5	q_2	0	0
3	q_6	q_3	0	0
4	q_7	q_4	0	0

2) Discrete Co-design: In the previous section, an 8-dimensional co-design results in a task-relevant constraint. We freeze a subset of 8 parameters to form the immutable hardware, while the rest are controllable during the task. Therefore, we can co-design 2^8 gripper fingers with varying performance and rendered control complexity based on the parameters chosen for control. We formulate an informed discrete search amongst these finger designs by plotting the individual task compatibility contributions of all the co-design parameters in Fig. 4a. We see that q_0 and q_1 are best for imparting tangential forces to the object, while q_2 and q_7 are adept at imparting normal forces.

We design four grippers by choosing at least one control parameter adept at producing forces in either direction. These designs and their task compatibilities, T as shown in Fig. 4b. The gripper where $\{q_1, q_2, q_3, q_4\}$ form the control parameters achieves the best task-compatibility index of 1.50. However, this comes at the cost of a high control complexity requiring four parameters. The gripper where $\{q_1, q_2\}$ are the control parameters has a close compatibility of 1.32 but a simpler control complexity requiring only two parameters. This experiment shows that n co-design parameters represent 2^n designs with varying performance and control complexities. We compare the individual contribution of each co-design parameter to the performance and select those best suited for the task to form the control parameters, whereas the rest form the fixed hardware. However, this requires us to balance the trade-off between the required performance and the rendered control complexity.

3) Effect of Initialization: In the final experiment, we evaluate the effect of initialization on our method. We randomly initialize at 100 points in the 8-dimensional co-design space and optimize using the methodology described in section IV-B.1. Fig. 4c shows the resulting task compatibilities as a function of two parameters, q_0 and q_1. We observe that while we get good task performance in reasonable co-designs, our gradient-based co-design method is susceptible to seeking out local maxima, making good initialization essential for a competent co-design solution. However, this is a common problem with gradient-based optimization.

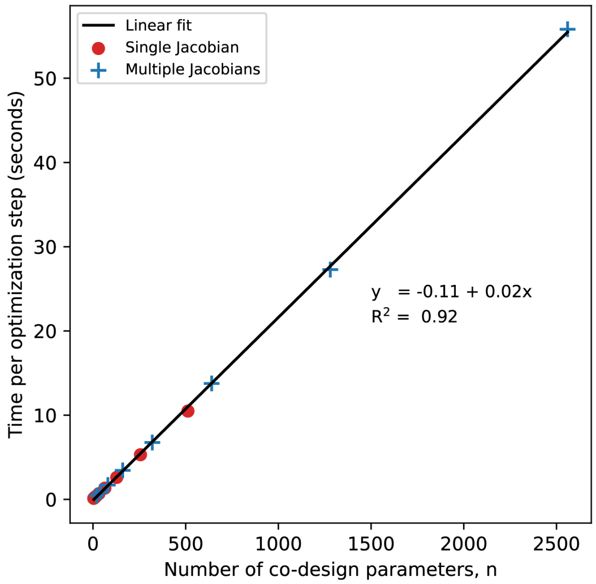

Fig. 5. Our optimization method using the co-design Jacobian scales well with increased co-design parameters, n. The time taken per optimization step scales linearly with n ($R^2 = 0.92$), using a single Jacobian (red) or multiple Jacobians (blue).

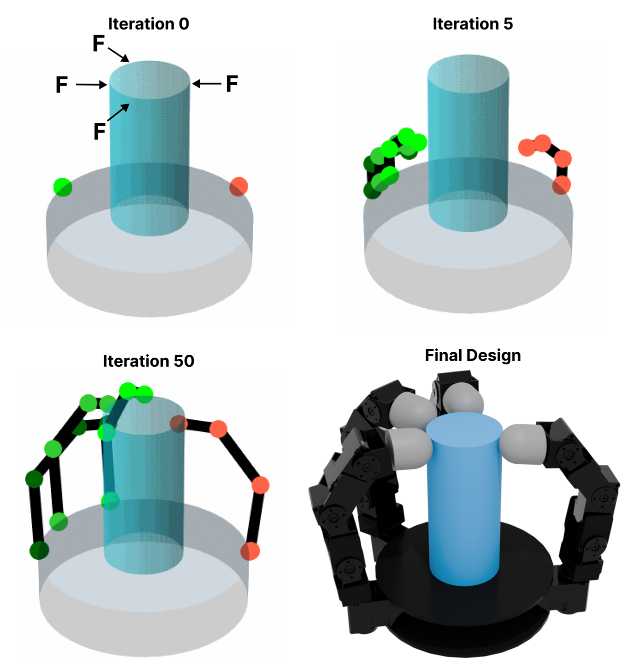

Fig. 6. High-dimensional co-designs are feasible when guided by constraints. We co-design an 80-dimensional multi-fingered hand guided by a precision grasp, exerting forces (F) at four points on the blue object. The green fingers are initialized with the same parameters, whereas the red finger initializes at a 120° angle from them. The figure shows the design progression during 50 iterations and the final hand design based on servo motors.

C. Multi-Fingered Hand Co-design

We extend our methodology to high-dimensional co-designs. Firstly, we examine the computational complexity of our method by increasing the number of co-design parameters, $n = \{4, 8, 16, 32, 64, 128, 256, 512\}$. We solve Eq. 1 for 100 steps on a single 3.50GHz CPU and measure the average time taken per optimization step. We form a $x \times n$ dimensional Jacobian where x is the task space dimension and equals 3. Gradient ascent using this Jacobian renders a complexity of n as shown by red markers in Fig. 5. Next, we use multiple Jacobians. We choose $n = \{20, 40, 80, 160, 320, 640, 1280, 2560\}$, resulting in $\frac{n}{20}$ number of $x \times 20$ dimensional Jacobians. Updating the co-design parameters sequentially using these Jacobians renders a linear time complexity, shown by blue markers in Fig. 5. Computing these multiple Jacobians in parallel can further improve this time complexity. Thus, our method scales well with increased dimensions of the co-design space.

Next, we use our methodology to co-design a multi-fingered hand. We initialize 80 co-design parameters equally distributed amongst 4 fingers. 3 fingers are initialized using the 20 parameters described in Table. I and the same initial configuration as in section IV-B.1. The thumb is initialized with $q_0 = 60°$, as shown by the red point on the palm in Fig. 6. Constraints formed during a precision grasp that impart large forces, F, in directions normal to the object at four points guide this co-design, shown in Fig. 6. The co-design parameters are updated for 50 steps using Eqn. (1). High design costs are added to the quality function to restrict changes in α_i or d_i that guide the optimization towards an anthropomorphic hand design. This co-design takes 86 seconds using a single $3.50Ghz$ CPU core. The hand design progression and a final hand design based on servo motors are shown in Fig. 6. This experiment shows that high-dimensional co-design of manipulation systems is feasible when guided by task-relevant constraints.

V. Conclusion

We present two conceptual tools for making high-dimensional co-design tractable. Firstly, we use environmental constraints (EC) as an underlying representation for co-design and show its effectiveness in structuring the co-design space for manipulation systems. Secondly, we present the co-design Jacobian that leverages the structure induced by ECs for efficient gradient-based optimization. These two tools were effective in co-designing diverse manipulation systems. Furthermore, the methodology scales well with increased parameters, making high-dimensional co-design feasible. Several directions can be pursued in the future based on the two basic principles described in this study. Firstly, this work can be supplemented by a methodology that provides reasonable initialization in the co-design space, essential for gradient-based optimization. Secondly, the structure rendered by ECs in the co-design space can also be verified using alternate quality functions for complex dynamic manipulation tasks. Lastly, this methodology can be extended to multi-objective tasks and manipulation in diverse environments.

References

[1] C. Eppner, R. Deimel, J. Álvarez Ruiz, M. Maertens, and O. Brock, "Exploitation of environmental constraints in human and robotic grasping," *The International Journal of Robotics Research*, vol. 34, no. 7, pp. 1021–1038, 2015.

[2] J.-H. Park and H. Asada, "Concurrent Design Optimization of Mechanical Structure and Control for High Speed Robots," in *American Control Conference*, 1993, pp. 2673–2679.

[3] K. S. Luck, H. B. Amor, and R. Calandra, "Data-efficient Co-Adaptation of Morphology and Behaviour with Deep Reinforcement Learning," *arXiv*, 2019.

[4] R. Deimel, P. Irmisch, V. Wall, and O. Brock, "Automated co-design of soft hand morphology and control strategy for grasping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, 2017, pp. 1213–1218.

[5] A. Zhao, T. Du, J. Xu, J. Hughes, J. Salazar, P. Ma, W. Wang, D. Rus, and W. Matusik, "Automatic Co-Design of Aerial Robots Using a Graph Grammar," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 11 260–11 267.

[6] N. Cheney, J. Bongard, V. SunSpiral, and H. Lipson, "Scalable co-optimization of morphology and control in embodied machines," *Journal of The Royal Society Interface*, vol. 15, 2018.

[7] J. Xu, T. Chen, L. Zlokapa, M. Foshey, W. Matusik, S. Sueda, and P. Agrawal, "An End-to-End Differentiable Framework for Contact-Aware Robot Design," in *Robotics: Science and Systems XVII*, 2021.

[8] K. Wampler and Z. Popović, "Optimal gait and form for animal locomotion," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 60:1–60:8, 2009.

[9] S. Ha, S. Coros, A. Alspach, J. Kim, and K. Yamane, "Computational co-optimization of design parameters and motion trajectories for robotic systems," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1521–1536, 2018.

[10] T. Dinev, C. Mastalli, V. Ivan, S. Tonneau, and S. Vijayakumar, "A Versatile Co-Design Approach For Dynamic Legged Robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 10 343–10 349.

[11] A. Gupta, S. Savarese, S. Ganguli, and L. Fei-Fei, "Embodied intelligence via learning and evolution," *Nature Communications*, vol. 12, no. 1, p. 5721, 2021.

[12] C. Schaff, A. Sedal, and M. R. Walter, "Soft Robots Learn to Crawl: Jointly Optimizing Design and Control with Sim-to-Real Transfer," *arXiv*, 2022.

[13] T. Chen, Z. He, and M. Ciocarlie, "Hardware as Policy: Mechanical and Computational Co-Optimization using Deep Reinforcement Learning," 2020.

[14] Y. Hu, J. Liu, A. Spielberg, J. B. Tenenbaum, W. T. Freeman, J. Wu, D. Rus, and W. Matusik, "ChainQueen: A Real-Time Differentiable Physical Simulator for Soft Robotics," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6265–6271.

[15] T.-H. Wang, P. Ma, A. E. Spielberg, Z. Xian, H. Zhang, J. B. Tenenbaum, D. Rus, and C. Gan, "SoftZoo: A Soft Robot Co-design Benchmark For Locomotion In Diverse Environments," 2023.

[16] Y. D. Zhong, J. Han, and G. O. Brikis, "Differentiable Physics Simulations with Contacts: Do They Have Correct Gradients w.r.t. Position, Velocity and Control?" 2022.

[17] J. Whitman and H. Choset, "Task-Specific Manipulator Design and Trajectory Synthesis," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 301–308, 2019.

[18] M. Toussaint, J.-S. Ha, and O. S. Oguz, "Co-Optimizing Robot, Environment, and Tool Design via Joint Manipulation Planning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6600–6606.

[19] R. Deimel and O. Brock, "Soft Hands for Reliable Grasping Strategies," in *Soft Robotics*, A. Verl, A. Albu-Schäffer, O. Brock, and A. Raatz, Eds. Springer Berlin Heidelberg, 2015, pp. 211–221.

[20] C. Eppner and O. Brock, "Planning Grasp Strategies That Exploit Environmental Constraints," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4947–4952.

[21] A. Bhatt, A. Sieler, S. Puhlmann, and O. Brock, "Surprisingly Robust In-Hand Manipulation: An Empirical Study," in *Robotics: Science and Systems XVII*, 2021.

[22] S. L. Chiu, "Task Compatibility of Manipulator Postures," *The International Journal of Robotics Research*, vol. 7, no. 5, pp. 13–21, 1988.